



PERMAINAN BERANIMASI UNTUK POCKET PC

oleh

NOORMAN BIN AZAHARI

di bawah penyeliaan

CIK SITI HAFIZAH HAMID

Perpustakaan SKTM

PROJEK LATIHAN ILMIAH TAHAP AKHIR

diserahkan kepada

FAKULTI SAINS KOMPUTER DAN TEKNOLOGI MAKLUMAT

Laporan Latihan Ilmiah yang dikemukakan kepada

FAKULTI SAINS KOMPUTER DAN TEKNOLOGI MAKLUMAT,

UNIVERSITI MALAYA

bagi memenuhi sebahagian dari keperluan

IJAZAH SARJANA MUDA SAINS KOMPUTER

**FAKULTI SAINS KOMPUTER
DAN TEKNOLOGI MAKLUMAT
UNIVERSITI MALAYA**

**PERMAINAN BERANIMASI
UNTUK POCKET PC
(WINDOWS CE 3.0)**

NOORMAN BIN AZAHARI

WEK 990299

WXES 3182 : LATIHAN ILMIAH II

PENSYARAH : CIK SITI HAFIZAH HAMID

MODERATOR : PN RAJA NOOR AINUN

**LAPORAN LATIHAN ILMIAH INI ADALAH UNTUK RUJUKAN
PELAJAR FSKTM YANG INGIN MENCEBURI BIDANG
PERMAINAN BERANIMASI (GAMING)**

ABSTRAK

Projek ini bertujuan untuk membangunkan satu program permainan yang boleh dimainkan dalam Pocket PC. Permainan ini boleh dimuatturunkan pada Pocket PC. Kaedah yang akan digunakan ialah kaedah prototaip dan menggunakan Microsoft eMbedded Visual Tools 3.0 sebagai medium pembangunan permainan beranimasi. Projek ini bermula pada 18 Mac 2002 dan tamat pada 6 September 2002. Projek ini merangkumi kajian mengenai Pocket PC dan pembangunan sistem permainan tersebut. Permainan tersebut hendaklah mempunyai animasi, menarik dan mudah difahami oleh pengguna Pocket PC.

PENGHARGAAN

Penghargaan “Assalamualaikum warahmatullahi wabarakaatuh”

Syukur Alhamdulillah dengan kuasa dan kebesaran Allah yang Maha Esa dan berkat selawat ke atas Rasulullah s.a.w, dapat juga saya melengkapkan sebahagian dari Projek Ilmiah Tahap 2 ini.

Di sini, saya ingin mengucapkan jutaan terima kasih serta penghargaan yang tak terhingga kepada pelbagai pihak yang terlibat dalam menjayakan projek permainan beranimasi ini. Saya ingin mengucapkan terima kasih kepada kedua ibu dan bapa saya atas sokongan yang diberikan sepenuhnya serta sumbangan kewangan yang ikhlas. Kedua, saya juga ingin mengucapkan terima kasih kepada Cik Siti Hafizah kerana telah banyak membantu saya dalam menjayakan projek ini. Segala nasihat dan tunjuk ajar yang diberikan tidak dapat saya lupakan. Tidak lupa juga ucapan terima kasih kepada Prof. Raja Noor Ainun selaku moderator saya di atas budi bicara puan dalam menilai projek ini.

Kepada seluruh sahabat dan saudara mara yang secara langsung atau tidak langsung membantu saya sepanjang projek ini dijalankan, diucapkan ribuan terima kasih. Segala jasa baik dari semua pihak amatlah saya hargai dan sanjungi hingga ke akhir hayat.

Noorman bin Azahari

WEK 990299

ISI KANDUNGAN

Penghargaan	i
Abstrak	ii
Senarai Gambarajah	vi

BAB 1

1.0	Pengenalan	1
1.1	Objektif Projek	2
1.2	Skop Projek	2
1.3	Draf Perancangan Projek dan Isi Kandungan Tesis ..	3

BAB 2

2.0	Pengenalan Kajian Literasi	6
2.0.1	Pencarian Maklumat	6
2.0.2	HCI (<i>Human-Computer Interaction</i>)	7
2.0.3	GUI (Rekabentuk Antaramuka Bergrafik).....	9
2.1	Kajian Permainan Beranimasi	11
2.1.1	Jenis-jenis Permainan Beranimasi untuk Pocket PC	11
2.1.2	Kesimpulan Daripada Analisa Permainan Beranimasi.....	14
2.2	Kajian Pocket PC	16
2.2.1	Fungsi Penggunaan Pocket PC.....	16
2.2.2	Teknologi terkini Pocket PC.....	17

BAB 3

3.0	Pengenalan Kajian Metodologi.....	20
3.1	Analisis Ringkas Metodologi.....	21
3.1.1	Pemprototaipan.....	21

BAB 3	3.2	Pemilihan Model.....	22
6.0	3.2.1	Kelebihan Model Prototap.....	23
6.1	3.2.2	Jenis-jenis Prototaip.....	23
	3.3	Model Prototaip Yang Dipilih.....	24
	3.3.1	Fasa-fasa Prototaip Permainan Beranimasi....	25
	3.4	Pemilihan Perisian.....	26
	3.5	Pemilihan Perkakasan.....	28
BAB 4			
	4.0	Pengenalan Rekabentuk.....	30
	4.1	Rekebentuk Keseluruhan Sistem Permainan Beranimasi....	31
	4.2	Rekabentuk Antaramuka Permainan.....	34
	4.3	Rekabentuk Pembangunan Sistem.....	35
BAB 5			
	5.0	Pengenalan Implementasi Sistem.....	37
	5.1	Persekitaran Pembangunan.....	37
	5.1.1	Keperluan Perkakasan.....	37
	5.1.2	Keperluan Perisian.....	38
	5.2	Pembangunan Sistem.....	40
	5.2.1	Peringkat Pembangunan.....	40
	5.3	Pengkodan Utama.....	44
	5.3.1	Pengkodan Permainan Beranimasi.....	44
	5.3.2	Pengkodan Game Library.....	45

BAB 6

6.0	Pengenalan Pengujian Sistem.....	47
6.1	Strategi Pengujian.....	47
6.1.1	Pengujian Unit.....	48
6.1.2	Pengujian Integrasi.....	48
6.1.3	Pengujian Sistem.....	49
6.1.4	Pengujian Penerimaan.....	50
6.2	Jenis Ralat.....	50
6.3	Penilaian Pengguna.....	51
6.4	Guna-Semula.....	52
6.4.1	Pengujian Black-Box.....	52
6.4.2	Pengujian White-Box.....	52

BAB 7

7.0	Penyelenggaraan Sistem.....	53
7.1	Jenis-jenis Penyelenggaraan.....	53
7.2	Dokumentasi.....	54
7.3	Penilaian Sistem.....	55
7.3.1	Kelebihan Sistem.....	55
7.3.2	Kekurangan Sistem.....	56
7.3.3	Harapan Pada Masa Hadapan.....	56

BAB 8

8.0	Masalah dan Penyelesaiannya.....	58
8.1	Kekuatan dan Perkembangan Projek.....	59

8.1 Kesimpulan Projek.....	60
Kod Sumber.....	61
Manual Pengguna dan Pentadbir.....	104
Rujukan.....	107
Rumusan.....	108

SENARAI GAMBARAJAH

Rajah 1 A.....	5
Rajah 2-A.....	11
Rajah 2-B.....	12
Rajah 2-C.....	13
Rajah 2-D.....	13
Rajah 2-E.....	14
Rajah 2-F.....	19
Rajah 3-A.....	22
Rajah 4-A.....	31
Rajah 4-B.....	32
Rajah 4-C.....	33
Rajah 4-D.....	34
Rajah 4-E.....	34
Rajah 4-F.....	35

Rajah 4-G.....36

Rajah 5-A.....40

Rajah 5-B.....46

1.1. PENGENALAN

Pocket PC merupakan salah satu perangkat portabel jenis alat elektronik jenis PDA (Personal Digital Assistant)[4] atau dengan kata lain adalah komputer tapak tangan. Kini, sudah sering di seluruh dunia menggunakan Pocket PC karena ia merupakan salah satu alat komunikasi yang mudah dipakai selain telepon genggam. Ia juga boleh berfungsi sebagai seperti komputer yang dipakai di rumah. Yang berbeda ialah saiz dan beberapa aplikasi tertentu. Model Pocket PC yang boleh didapati di pasaran masa ini antaranya ialah Casopole 334-500.

BAB 1

PENGENALAN

Salah satu jenis peranti komunikasi untuk Pocket PC adalah telefon genggam. Ia boleh digunakan dengan telefon genggam atau peranti elektronik yang lain. Kini, kebanyakan peranti Pocket PC menjadi peranti komputer peribadi juga. Oleh itu, dalam Pocket PC, terdapat di dalam projek ini, akan dihasilkan dengan lebih banyak kegunaan untuk memudahkan peranti peramiran untuk Pocket PC agar menjadi peranti untuk peranti yang lain.

1.0 PENGENALAN

Pocket PC merupakan salah satu daripada pelbagai jenis alat elektronik jenis PDA (*Personal Digital Assistant*)[4] atau dengan nama lain ialah komputer tapak tangan. Kini, ramai orang di seluruh dunia menggunakan Pocket PC kerana ia merupakan salah satu alat komunikasi yang mudah dipakai selain telefon bimbit. Ia juga boleh berfungsi hampir seperti komputer yang dipakai di rumah. Yang berbeza ialah saiz dan beberapa aplikasi tertentu. Model Pocket PC yang boleh didapati dipasaran masa kini di antaranya ialah Cassiopeia EM-500, Jornada 540 series dan Compaq iPAQ H3600.

Biasanya, pengguna Pocket PC menggunakannya sebagai catatan temujanji dan menyimpan maklumat penting yang perlu di bawa ke mana-mana sahaja. Tetapi, ada juga aplikasi Pocket PC yang menyimpan program permainan memandangkan kapasitinya yang lebih besar daripada telefon bimbit. Malah, permainan komputernya lebih baik daripada telefon bimbit.

Namun begitu, pembinaan permainan untuk Pocket PC adalah lebih kompleks berbanding dengan telefon bimbit dan beberapa alat permainan elektronik yang lain. Kini, pembangun permainan Pocket PC mencuba agar permainan komputer peribadi juga boleh dimainkan di dalam Pocket PC. Tetapi, di dalam projek ini, akan diterangkan dengan lebih mendalam bagaimana untuk membangunkan permainan beranimasi untuk Pocket PC agar menjadi panduan untuk pembangun yang lain.

1.1 OBJEKTIF PROJEK

- i) Mengenalpasti aplikasi dan program yang perlu untuk membangunkan permainan beranimasi.
- ii) Mengkaji pengaturcaraan yang digunakan dalam membangunkan permainan beranimasi.
- iii) Merekebentuk dan membina satu permainan beranimasi yang menarik dan mudah difahami oleh pengguna.
- iv) Memberikan kepuasan kepada pengguna apabila bermain permainan beranimasi dari Pocket PC.

1.2 SKOP PROJEK

Projek ini memerlukan skop projek sebagai hala tuju untuk melengkapkan projek ini. Skop projek ini merangkumi pengguna dan pentadbir :

i) pengguna

- boleh bermain permainan yang dicipta di dalam Pocket PC dengan menggunakan panduan yang diberikan.

ii) pentadbir

- memastikan bahawa permainan yang dicipta dapat dimuatturunkan pada Pocket PC.
- menyediakan paparan dan panduan untuk pengguna bermain permainan tersebut.

1.3 Draf Perancangan Projek dan Isi Kandungan Tesis

Projek ini dimulakan dengan mengenalpasti masalah yang boleh wujud dengan lebih terperinci. Pengumpulan maklumat dan kaji selidik dilakukan sebagai panduan untuk melengkapkan projek. Maklumat yang dikumpulkan bukan sekadar kandungan projek malah mengenai perkakasan dan perisian yang akan digunakan sebagai alat pembangunan sistem.

Berikut merupakan penerangan ringkas mengenai bab-bab yang terkandung di dalam projek ini :

Bab 1 – Pengenalan

Bab ini menerangkan serba sedikit mengenai Pocket PC dan permainan beranimasi yang wujud ketika ini. Selain itu, ia mengandungi objektif projek, skop projek, jadual perancangan dan draf kandungan tesis.

Bab 2 – Analisis Literasi

Mengandungi objektif literasi, sedikit penerangan mengenai pembikinan antaramuka, penerangan mengenai Pocket PC (sejarah, fungsi, teknologi) dan penerangan mengenai permainan beranimasi (system, rekabentuk, pengaturcaraan).

Bab 3 – Analisis Metodologi

Penerangan secara lengkap mengenai kaedah yang akan digunakan untuk membangunkan sistem, cara dan bagaimana ia hendak diimplementasikan. Perbincangan masalah dan cara penyelesaiannya jika ada dengan berpandukan metodologi yang dipilih.

Bab 4 – Rekabentuk Sistem Permainan

Merekabentuk permainan beranimasi dan antaramukanya berpandukan metodologi yang dipilih dan simulator yang ada.

Bab 5 – Implementasi Permainan

Menerangkan bagaimana sistem dibangunkan menggunakan simulator, perkakasan dan perisian yang disediakan. Permainan yang dibuat mestilah bersesuaian dengan kapasiti Pocket PC.

Bab 6 – Pengujian Sistem Permainan

Bagaimana sistem diuji dan kaedah pengujian yang akan digunakan. Kaedah pengujian dan hasilnya didokumentasikan

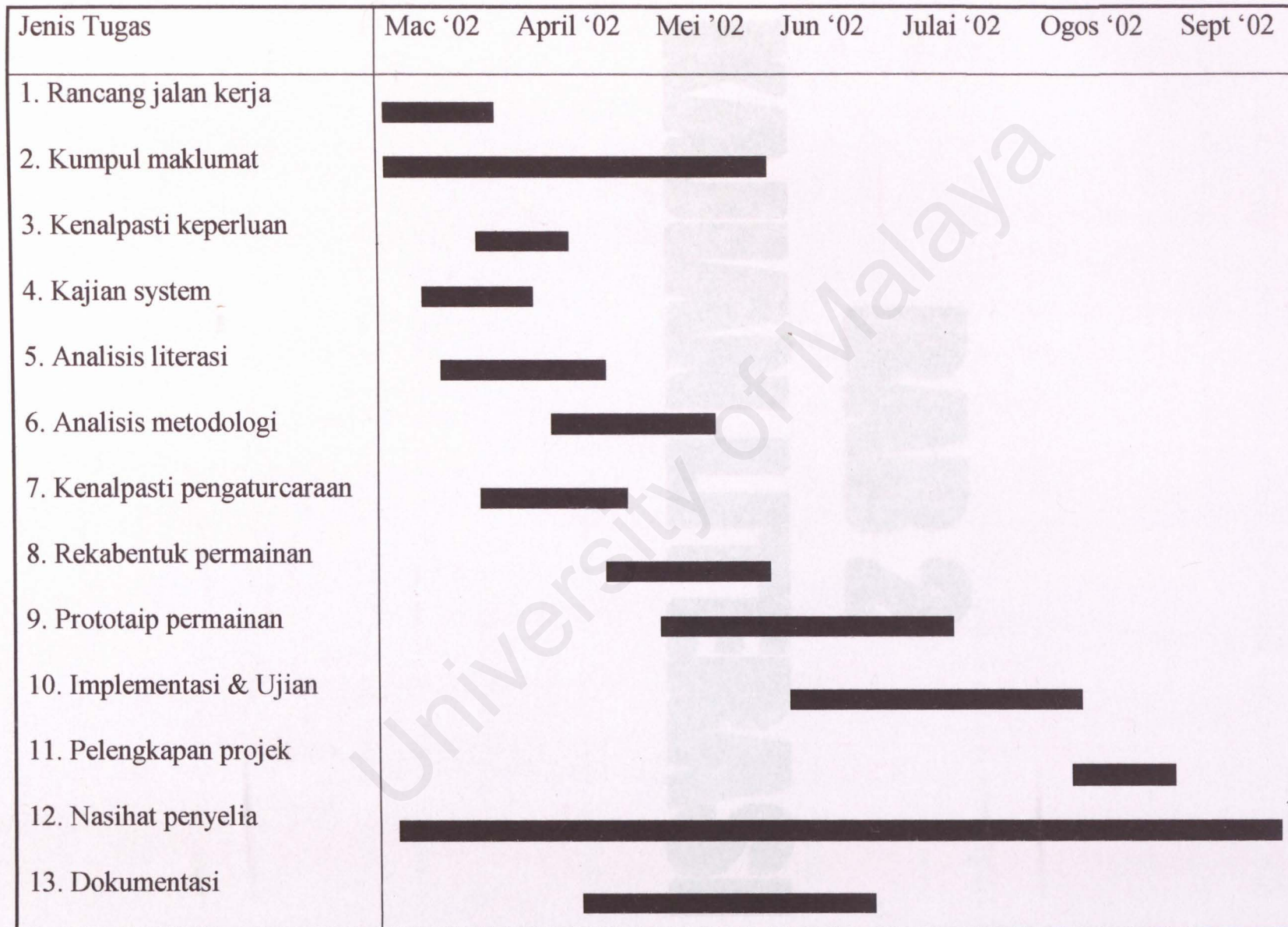
Bab 7 – Penyelenggaraan dan Penilaian Sistem

Permainan yang telah siap dinilai dan sebarang komen atau perubahan dilakukan ketika ini untuk mengelakkan sebarang kesilapan. Fungsian tambahan juga diterima bagi meningkatkan kualiti sistem.

Bab 8 – Masalah dan Kesimpulan

Merupakan penilaian terakhir, pengajaran yang boleh didapati dan harapan untuk masa depan.

Rajah 1 A : Carta Gannt Pembangunan Permainan untuk Pocket PC



3.4 PENGENALAN KAJIAN LITERASI

Kajian literasi dibuat untuk meningkatkan pemahaman mengenai system yang berlaku menggunakan ia terbahagi kepada 3 komponen:

- penerapan teknologi
- analisis maklumat
- aspek of kesedaran yang dimiliki

Ketiga-tiga komponen ini akan berkesan jika ia dapat menghasilkan perubahan yang signifikan.

Manfaat:

3.4.1 Peningkatan Literasi

Untuk mewujudkan perubahan, maka ia memerlukan para pemertua

BAB 2 KAJIAN LITERASI

3.4.2 Literasi Digital

• Literasi digital adalah keupayaan untuk mencari, memahami, menilai dan menggunakan maklumat digital.

• Literasi digital adalah keupayaan untuk mencari, memahami, menilai dan menggunakan maklumat digital.

3.4.3 Literasi Media

• Literasi media adalah keupayaan untuk memahami dan menilai maklumat yang datang dari pelbagai sumber. Contohnya:

• buku, majalah, surat, radio, televisyen, internet, blog, dokumen dan majalah.

3.4.4 Literasi

• Literasi adalah keupayaan untuk mencari, memahami, menilai dan menggunakan maklumat yang datang dari pelbagai sumber.

• Literasi adalah keupayaan untuk mencari, memahami, menilai dan menggunakan maklumat yang datang dari pelbagai sumber.

• Literasi adalah keupayaan untuk mencari, memahami, menilai dan menggunakan maklumat yang datang dari pelbagai sumber.

• Literasi adalah keupayaan untuk mencari, memahami, menilai dan menggunakan maklumat yang datang dari pelbagai sumber.

2.0 PENGENALAN KAJIAN LITERASI [2]

Kajian literasi dibuat untuk meningkatkan pemahaman mengenai system yang hendak dibangunkan. Ia terbahagi kepada 3 komponen :

- pencarian maklumat
- analisis maklumat
- sistesis @ kesimpulan yang dibuat

Ketiga-tiga komponen ini akan diproses dan dikaji untuk panduan dalam membangunkan sistem.

2.0.1 Pencarian Maklumat

Untuk membina permainan untuk Pocket PC ini, beberapa cara pencarian maklumat digunakan untuk membantu dalam melengkapkan projek. Antara cara yang digunakan ialah :

1. bahan dari penyelia
 - pelbagai bahan bacaan disediakan oleh penyelia Cik Siti Hafizah dan panduan dari tesis yang lalu.
2. buku dan majalah
 - pelbagai bahan bacaan dikumpul dari pelbagai sumber. Contohnya buku dipinjam dari perpustakaan, bilik dokumen dan majalah.
3. internet
 - pelbagai laman internet dilayari untuk mencari maklumat yang berkaitan dengan projek memandangkan ada maklumat yang tidak dapat dicari dari buku dan majalah. Contoh laman web yang dilayari ialah www.Microsoft.com dan www.PrimaGameDev.com.

2.0.2 HCI (Human Computer Interaction) [2]

HCI boleh didefinisikan sebagai :

"Human-computer interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them "

Ia merupakan satu jenis interaksi antara komputer dengan manusia yang menitikkan rekabentuk, penilaian dan implementasi antara sistem komputer dan manusia. Ia mengkaji keadaan sekeliling yang mempengaruhi interaksi antara mereka.

Fokus utama HCI adalah interaksi antara satu atau lebih manusia dengan mesin. Skop HCI bukanlah untuk komputer sahaja tetapi lebih menyeluruh kerana antaramuka bukan sahaja pada skrin komputer tapi ada juga pada ketuhar gelombang, telefon bimbit, kawalan jarak jauh dan lain-lain lagi. Interaksi antara manusia dan komputer menjadi lebih efektif jika wujud satu 'persefahaman' di mana pengguna tidak perlu membaca manual sebelum menggunakan peralatan.

Untuk menjelaskan lagi tentang interaksi manusia dan komputer, ia boleh diterangkan sebagai :

- HCI menitikberatkan kepada hasil yang akan dicapai bila interaksi antara komputer dan manusia berjaya.
- struktur komunikasi antara manusia dan komputer.
- kebolehan manusia menggunakan antaramuka.
- algoritma dan pengaturcaraan untuk membentuk antaramuka itu sendiri.

- faktor yang timbul apabila merekabentuk dan membangunkan antaramuka tersebut.
- proses spesifikasi, rekabentuk dan implemetasi antaramuka.

Kesimpulannya, HCI boleh diterangkan secara kasar seperti berikut :

a) HCI asli

- model meta HCI.

b) Penggunaan mesin

- organisasi sosial manusia dan kerja yang harus diselesaikan.
- ruang aplikasi.
- kebolehan manusia dan komputer.

c) ciri-ciri manusia

- cara pemprosesan maklumat manusia.
- penggunaan bahasa, komunikasi dan interaksi.
- ergonomik.

d) senibina antaramuka dan system komputer itu sendiri

- peranti input dan output.
- teknik dialog yang digunakan.
- jenis dialog¹.
- penggunaan grafik komputer.
- senibina dialog.

e) proses pembangunan

- pendekatan pembangunan.
- teknik implementasi.

¹ Dialog yang dimaksudkan dalam senibina antaramuka adalah bagaimana interaksi manusia dan komputer

- teknik penilaian keseluruhan.
- contoh system dan kajian kes.

f) paparan projek dan penilaian

HCI adalah satu pendekatan yang melibatkan 'kerjasama' antara pengguna dan komputer. HCI akan digunakan sepanjang analisa projek ini untuk memastikan ada sefahaman antara sistem ini dan pengguna yang akan menggunakannya. Dengan menggunakan teknik HCI, antaramuka yang terhasil mudah difahami dan senang digunakan.

2.0.3 GUI (rekabentuk antaramuka bergrafik) [1]

Rekabentuk antaramuka bergrafik adalah media komunikasi antara manusia dan komputer. Antaramuka adalah satu bahagian komputer yang membenarkan manusia berinteraksi dengan komputer. Antaramuka yang selalu digunakan adalah :

- ◆ perkakasan komputer seperti papan kekunci, tetikus, skrin dan lain-lain lagi.
- ◆ imej yang dijanakan oleh perisian seperti Windows, menu, mesej dan sebagainya.
- ◆ dokumentasi pengguna, selalunya benda bercetak seperti manual pengguna dan kad rujukan.

Pengguna tiada capaian ke bahagian dalaman komputer kecuali melalui antaramuka. Tujuan utama antaramuka ialah kebolehgunaan sistem oleh pengguna dan paling kritikal adalah antaramuka yang boleh digunakan. Kebolehgunaan termasuk kesenangan untuk belajar dan keberkesanan menggunakannya. Ini merupakan faktor kualiti terpenting yang susah dicapai. Kebolehgunaan bermaksud penyesuaian manusia yang menggunakan

sistem dengan sistem itu sendiri. Ini akan menimbulkan isu psikologi yang rumit terhadap ingatan manusia, persepsi dan tanggapan konseptual.

Untuk merekabentuk antaramuka, tiga konsep utama perlu diikuti iaitu :

1. Pengguna dapat mengawal keadaan

- interaksi didefinisikan dengan cara tidak menyukarkan pengguna.
- menghasilkan interaksi yang fleksibel.
- pengguna dapat menguasai keadaan dengan interaksi.
- langkah teknikal disembunyikan dari pengguna.
- interaksi mestilah paparan secara terus di atas skrin.

2. Pastikan pengguna tidak perlu mengingat

- memberikan panduan untuk setiap langkah.
- paparan antaramuka mestilah bermakna.

3. antaramuka haruslah konsisten

- membenarkan pengguna bermain dalam konsep yang bermakna.
- meneruskan konsistensi di sepanjang sistem.
- tidak menukar gaya antaramuka yang biasa digunakan oleh pengguna kecuali ia benar-benar perlu.

Tiga konsep utama di atas adalah asas kepada prinsip rekabentuk antaramuka dalam membangunkan antaramuka yang berkualiti.

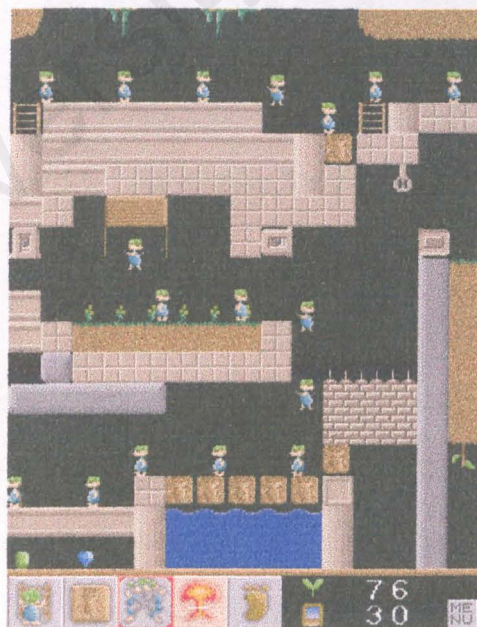
2.1 KAJIAN PERMAINAN BERANIMASI

Terdapat 2 jenis permainan beranimasi untuk Pocket PC ini iaitu permainan perseorangan dan rangkaian. Permainan perseorangan biasanya diprogram untuk jenis seorang individu seperti permainan kad atau papan. Permainan rangkaian pula adalah untuk dua atau lebih individu yang disambungkan menggunakan kabel atau infrared. Berikut adalah genre-genre permainan beranimasi yang ada untuk Pocket PC :

2.1.1 Jenis-jenis Permainan Beranimasi untuk Pocket PC[5]

1. Permainan Aksi dan Arked

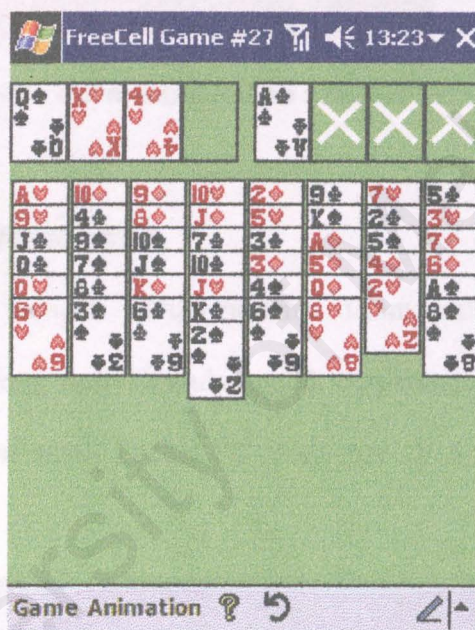
Genre ini biasanya pelbagai, seperti permainan klasik seperti *Pac-Man*, menembak seperti *R-Type* dan perlawanan seperti *Mortal Kombat*. Permainan jenis ini biasa boleh didapati dalam bentuk 2-dimensi atau 3-dimensi. Rajah di bawah adalah contoh permainan jenis arked dan aksi.



Rajah 2-A : Permainan Arked dan Aksi – *Isitar*

2. Permainan Kad, Papan dan Teka-teki

Permainan jenis ini merupakan yang paling popular kerana ia paling mudah untuk dibangunkan pada pelantar yang baru. Ia merupakan permainan yang banyak menggunakan pemikiran dan kebijaksanaan. Ada pelbagai permainan kad, papan dan teka-teki seperti *PalmChess*, *Dominoes* dan *Crossword*. Rajah di bawah adalah contoh permainan kad, papan dan teka-teki.



Rajah 2-B : Permainan Kad – *Pocket Freecell*

3. Permainan Sukan

Antara jenis permainan yang amat diminati oleh semua orang kerana hampir menyerupai permainan sebenar, cuma berbeza ia dimainkan dalam komputer. Ada pelbagai jenis sukan yang diambil fungsinya untuk dijadikan sebagai permainan beranimasi seperti bola sepak, ping-pong, tennis, golf dan lain-lain. Contohnya ialah *ZIOGolf*, *Pocket Football* dan *Pong*.



Rajah 2-C : Permainan Sukan : *Pocket Tennis*

4. Permainan Strategi Masa-Nyata

Permainan genre ini menjadi pilihan pertama dalam dunia 'gaming' masa ini. Cara bermain seperti merancang satu kumpulan untuk berperang dengan kumpulan lain. Contoh permainan strategi ini untuk Pocket PC ialah *Age Of Empires*, *Battlefield* dan *Galactics Realms*.



Rajah 2-D : Permainan Strategi : *Age Of Empires*

5. Permainan Simulasi

Ini juga merupakan antara permainan beranimasi yang menarik perhatian ramai orang kerana cara bermain umpama sedang memandu kereta dan kapal terbang. Contoh permainan simulasi yang popular ialah *Flight Simulator*, *RallyCE* dan *WingCommander*.



Rajah 2-E : Permainan Simulasi : *Wing Commander*

2.1.2 Kesimpulan Daripada Analisa Permainan Beranimasi

Terdapat pelbagai jenis lagi permainan beranimasi yang boleh didapati dalam Pocket PC ini. Namun begitu, permainan beranimasi untuk Pocket PC ini masih belum dapat menandingi kehebatan permainan untuk komputer peribadi kerana Pocket PC masih belum mempunyai keupayaan untuk melarikan permainan 3-dimensi yang lebih canggih. Contohnya, *DirectX*[11] adalah untuk keupayaan grafik dan audio untuk komputer peribadi tetapi untuk Pocket PC adalah *GAPI (Game Application Programming Interface)*[6] yang berkeupayaan lebih rendah dari *DirectX*.

Semua contoh permainan beranimasi untuk Pocket PC yang dinyatakan di atas boleh didapati pada laman web <http://www.jimmy.com> . Terdapat ramai pembangun permainan beranimasi untuk Pocket PC yang menghasilkan permainan-permainan ini sejak dari tahun 70-an lagi. Namun, tidak ramai pembangun permainan beranimasi ini berasal dari Malaysia dan biasanya mereka berasal dari USA , Jepun dan Jerman. Pembangun permainan beranimasi ini hendaklah pakar dalam bahasa pengaturcaraan, rekabentuk grafik, rekabentuk antaramuka dan audio. Ini merupakan asas untuk membangunkan permainan beranimasi yang berkualiti dan menarik.

Kesimpulannya, memandangkan permainan beranimasi untuk Pocket PC ini hanya memerlukan kapasiti pembinaan yang kecil, maka pengguna juga dapat membangunkan permainan beranimasinya sendiri kerana hanya pengetahuan asas mengenai pengaturcaraan boleh membangunkan permainan beranimasi yang paling ringkas.

2.2 KAJIAN POCKET PC[4]

2.2.1 Fungsi Penggunaan Pocket PC

1. Sistem Pengendalian (*Operating System*)

- Pocket PC dilarikan menggunakan Windows CE 3.0 (*"Windows Powered" Pocket PC*)

- Menyokong operasi system 32-bit

2. Aplikasi

- Terbahagi kepada dua memori iaitu ROM (16MB) dan RAM
- ROM – *Windows CE O/S, Media Player, Pocket Word, Pocket Excel, Peocket IE dan Pocket Reader.*
- RAM – *Object Store, My Documents, Audible Audiobooks, Reader eBooks, AvantGo Web Pages dan Games.*

3. Peralatan Input

- Beberapa butang arah untuk memilih aplikasi
- Butang *on/off*
- Pen Stylus – untuk menulis secara elektronik ke atas skrin atau memilih aplikasi
- Tidak mempunyai papan kekunci tetapi skrin memaparkan bentuk papan kekunci untuk digunakan bersama pen *Stylus.*

4. Keperluan Perkakasan

- Pemproses (CPU) – mempunyai 3 prosesor biasa dipakai iaitu *Hitachi SH3, NEC VR (MIPS)* dan *Intel StrongARM.* Masing-masing dengan kelajuan 150MHZ.
- Paparan Skrin – mempunyai pelbagai warna dan biasanya jenis *active matrix* atau *passive.*
- Memori – mempunyai kapasiti sebanyak 8MB, 16MB, 32MB dan 64MB.

- Audio – menyokong audio stereo 16-bit
- Spesifikasi – 150MHz prosesor, paparan warna 12-bit, 32MB memori utama, slot tambahan (CF, MMC dll) dan 'headphone'.

2.2.2 Teknologi Terkini Pocket PC

Pocket PC telah dicipta pada pertengahan 90'an, pada peringkat awal Internet diperkenalkan. Idea penggunaannya dibangunkan oleh Microsoft dengan menggunakan system pengendalian Windows CE untuk perhubungan tanpa wayar. Windows CE merupakan perisian yang digunakan sebagai pelantar pada perkakasan yang dipanggil PDA (*Personal Digital Assistant*).

Beberapa teknologi telah dikenalpasti untuk meningkatkan lagi penggunaan Pocket PC di kalangan masyarakat terutamanya dalam aspek permainan. Di bawah adalah sedikit penerangan mengenai teknologi terkini yang telah dibangunkan :

1. Servis Tanpa Wayar (*Wireless Services*)[7]

- Terdapat pilihan untuk sambungan Internet tanpa wayar setelah teknologi telefon digital dan selular diperkenalkan.
- Beberapa pengeluar perkakasan tanpa wayar menyahut seruan asas IEEE untuk meningkatkan pengeluaran.
- Teknologi tanpa wayar terbahagi 2 iaitu Ethernet tanpa wayar "jarak-pendek" dan servis Internet "jarak-jauh".

2. Sistem Selular (*Cellular Systems*)[8]

- Menggunakan sambungan system digital dan analog.
- Biasanya, system digital jauh lebih baik dari system analog tetapi ada perkakasan yang tidak boleh menggunakan system digital.

3. Ethernet Tanpa Wayar (*Wireless Ethernet*)[7]

- Merangkumi *LAN (Local Area Network)*.
- Kini sudah ada untuk dipasang pada komputer peribadi.
- Memerlukan kad host disambungkan ke hub rangkaian ke komputer peribadi.

4. *Ricochet*[7]

- Salah satu pilihan terbaik untuk *WAN (Wide Area Network)* / hubungan Internet tanpa wayar.
- Kuasa rangkaian sebanyak 128kbps.

5. *BlueTooth*[8]

- teknologi radio berkuasa rendah yang menyambung perkakasan elektronik tanpa menggunakan kabel.
- data dan suara boleh ditukar dalam jarak 10 meter tanpa perlu menyusun perkakasan ini.

Rajah 2-F : Contoh Pocket PC Masa Kini



3.0 PENGENALAN KAJIAN METODOLOGI

Metodologi ialah kaedah atau prosedur yang akan digunakan untuk melaksanakan sesuatu sistem, program atau projek. Tujuannya ialah untuk membantu dalam bagaimana sistem itu hendak dilaksanakan dan memberi kemudahan kepada pembangunan sistem untuk menunjukkan bagaimana sistem tersebut berfungsi.

Metodologi yang akan digunakan mestilah memenuhi ciri-ciri pembangunan sistem dalam pembangunan sistem yang berkaitan. Di sini, segala sesuatu yang berkaitan dengan pembangunan sistem akan dijelaskan dalam bab ini.

BAB 3 METODOLOGI

3.0 PENGENALAN KAJIAN METODOLOGI

Maksud metodologi ialah kaedah atau prosedur yang akan digunakan untuk membangunkan sesuatu sistem, program atau projek. Tujuannya ialah untuk membuat andaian bagaimana sistem itu hendak dibangunkan dan memberi kemudahan kepada pembangun sistem untuk menerangkan bagaimana sistem tersebut berfungsi.

Metodologi yang akan digunakan mestilah memenuhi ciri-ciri pembangunan sistem dalam menghasilkan suatu sistem yang berkualiti. Di sini, segala kesilapan atau masalah awal dapat dikesan untuk mengurangkan masalah yang akan timbul dalam fasa-fasa seterusnya. Selain itu, ia juga mendefinisikan fasa-fasa dalam pembangunan projek dan memberi panduan terhadap pengurusan dan pengawalan projek.

Metodologi yang baik memberi banyak kelebihan kepada pembangunan sistem. Di antara kelebihan yang boleh dicapai dengan menggunakan metodologi yang terbaik ialah :

- memberikan satu piawaian untuk pembangun projek agar tidak risau apa yang perlu dilakukan dan yang telah dilakukan.
- setiap satu fasa akan menghasilkan hasil yang maksimum dalam pembangunan setiap fasa.
- penyemakan semula akan menjadi lebih mudah jika setiap prosedur diikuti dengan teliti.
- peningkatan kualiti sistem yang dibangunkan dengan memberikan panduan yang perlu dalam setiap fasa.
- memberikan pemahaman yang lebih dalam mengesahkan keperluan pengguna.
- memudahkan pihak pengurusan projek membuat semakan perkembangan projek dan senarai semak tugas yang telah dilakukan.

- memberikan kemudahan merancang dan mengawal projek.

Metodologi yang baik mestilah mempunyai ciri-ciri yang berikut :

- senang digunakan dan difahami oleh penganalisa dan pengaturcara.
- merangkumi semua fasa dalam pembangunan sistem.
- berkaitan dengan aplikasi yang akan dibangunkan.
- dokumentasi yang berkualiti disediakan.

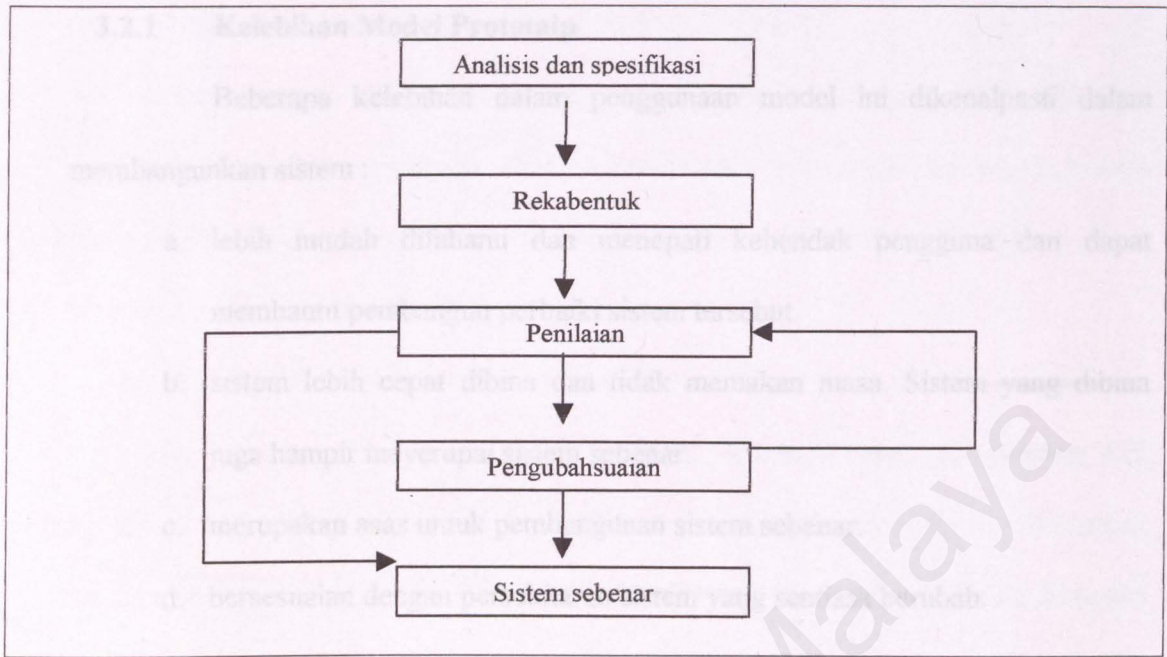
3.1 ANALISIS RINGKAS METODOLOGI

Di bawah merupakan beberapa penerangan secara ringkas metodologi yang dipilih yang telah dikenalpasti boleh digunakan untuk membangunkan projek permainan beranimasi untuk Pocket PC.

3.2.1 Pemprototaipan[3]

Makna prototaip ialah sistem percubaan berdasarkan spesifikasi dan keperluan yang telah dikaji dengan mendalam tapi masih belum disahkan, diimplimentasikan dan diselenggarakan. Merupakan pendekatan berasaskan evolusi terhadap pembangunan perisian. Rekabentuk dan keperluan dikaji berulang kali untuk mengurangkan risiko dan ketidakpastian dalam pembangunan sistem.

Terdapat beberapa langkah penting dalam membangunkan sistem menggunakan model ini iaitu analisis dan spesifikasi, rekabentuk, penilaian dan pengubahsuaian :



Rajah 3 A : Model Pemprototaipan (*Prototyping*)

3.4 PEMILIHAN MODEL

Model yang akan digunakan untuk membangunkan permainan beranimasi ini ialah model prototaip memandangkan ia dapat menyelesaikan beberapa kekurangan yang terdapat dalam model air terjun dan model V. Satu permainan beranimasi didirikan dahulu dengan menggunakan segala keperluan, maklumat dan berpandukan spesifikasi yang telah dikumpulkan. Kemudian, segala masalah atau pembetulan yang timbul dapat dikenalpasti dan prototaip sistem dapat dibetulkan dan diubahsuai mengikut citarasa pengguna. Objektifnya di sini ialah untuk mengurangkan risiko dan ketidakpastian yang mungkin muncul dalam pembangunan sistem.

3.2.1 Kelebihan Model Prototaip

Beberapa kelebihan dalam penggunaan model ini dikenalpasti dalam membangunkan sistem :

- a. lebih mudah difahami dan menepati kehendak pengguna dan dapat membantu pembangun memperbaiki sistem tersebut.
- b. sistem lebih cepat dibina dan tidak memakan masa. Sistem yang dibina juga hampir meyerupai sistem sebenar.
- c. merupakan asas untuk pembangunan sistem sebenar.
- d. bersesuaian dengan persekitaran sistem yang sentiasa berubah.
- e. dapat dikembalikan ke fasa tertentu jika terdapat kesilapan dalam fasa tersebut untuk pengubahsuaian.
- f. menggambarkan aplikasi sebenar yang akan diimplementasikan.
- g. pengguna terlibat dalam fasa permulaan pembangunan sistem.
- h. fleksibel mengikut kehendak pengguna.

3.2.2 Jenis-jenis Prototaip[9]

Banyak projek yang menggunakan model ini untuk membangunkan sistem. Namun, hasil akhir projek adalah berlainan dan komen atau cadangan pengguna juga adalah berbeza. Maka, wujudlah pelbagai versi baru dalam mendefinisikan model prototaip. Terdapat 3 jenis pendekatan dalam model ini untuk membangunkan sistem :

- Prototaip *evolutionary* atau *rapid*

Satu sampel sistem dibangun segera tanpa sebarang maklumat dari pengguna. Kemudian, pengguna diminta berinteraksi dengannya dan segala komen dan informasi dari pengguna dikumpulkan sebagai panduan lengkap untuk membangunkan sistem yang lebih baik.

- Prototaip *throw away*

Prototaip yang tidak langsung memenuhi kehendak pengguna dibuang atau diabaikan kerana masa dan kos pembaikan adalah tinggi. Ini bermakna kehendak sebenar pengguna telahpun dikenalpasti dan prototaip yang dibuang tidak dianggap membazir.

- Prototaip *incremental*

Merupakan gabungan prototaip *rapid* dan model air terjun. Sistem dibangun secara berasingan dalam tempoh masa yang berlainan. Satu rekebentuk diwujudkan untuk keseluruhan sistem tetapi dibangun dalam pecahan kecil.

3.3 Model Prototaip yang Dipilih

Untuk membangunkan permainan beranimasi ini, kaedah pemprototaipan pantas akan digunakan di mana sejenis permainan akan dibangun dan beberapa orang pengguna akan berinteraksi secara terus dengan sistem tersebut. Kemudian, pengguna akan memberikan cadangan, komen dan sebarang maklumat untuk meningkatkan kualiti permainan. Tujuannya ialah hendak mendapatkan jenis permainan yang dapat memenuhi kehendak dan citarasa pengguna malah boleh membantu membentuk permainan yang lebih menarik dan sesuai.

Faktor-faktor pemilihan model prototaip ini adalah seperti berikut :

1. menitikberatkan kehendak pengguna di mana mereka mencuba dahulu permainan yang dicipta dan memberikan maklumbalas yang diperlukan.
2. spesifikasi sistem yang dibangunkan adalah mengikut keperluan sistem sahaja.
3. sesetengah sistem telah dibangunkan semasa pembangunan prototaip.
4. implementasi adalah berdasarkan kepada sebahagian model yang berfungsi sahaja.

3.3.1 Fasa-fasa Prototaip Permainan Beranimasi

Untuk membangunkan permainan beranimasi untuk Pocket PC ini, sistem ini dibahagikan kepada 5 fasa :

1) Fasa Perancangan

- mengenalpasti objektif dan keperluan permainan.
- mengumpul maklumat dan spesifikasi Pocket PC.
- analisis semua maklumat permainan, Pocket PC, perkakasan dan perisian yang akan digunakan.

2) Fasa Rekabentuk

- merekabentuk sistem permainan beranimasi yang sesuai untuk pelantar Pocket PC.
- merekabentuk antaramuka sistem untuk kemudahan pengguna.
- merekabentuk pangkalan data.
- mendirikan carta dan pelan untuk rekabentuk yang telah dibuat.

3) Fasa Pembangunan

- membuat permainan beranimasi menggunakan perisian yang dipilih berpandukan maklumat yang disediakan.
- menyediakan pangkalan data untuk memuatkan permainan.
- mencipta antaramuka untuk interaksi permainan dengan pengguna.

4) Fasa Implementasi dan Penilaian

- pengkodan permainan beranimasi ke simulator Pocket PC.
- pembetulan demi pembetulan dilakukan untuk mencapai hasil akhir yang diharapkan.
- hasil akhir diuji dan dinilai menggunakan beberapa kaedah dan pendekatan.

5) Fasa Interaksi Pengguna

- sistem yang lengkap didedahkan ke pengguna untuk dicuba.
- manual pengguna tentang permainan disediakan.

3.4 PEMILIHAN PERISIAN[5]

Berikut adalah beberapa perisian yang akan digunakan untuk membangunkan permainan beranimasi untuk Pocket PC :

i. Microsoft eMbedded Visual Basic 3.0 (EVB)

Hampir serupa dengan VBA untuk komputer biasa tetapi ini untuk pembangunan aplikasi Pocket PC. EVB berbeza dengan VBA kerana terdapat beberapa fungsi yang tidak disokong. Jadual di bawah menerangkan perbezaan antara EVB dan VBA :

Sokongan Fungsi	VBA	EVB
Intrinsic Variable Types (long, string etc)	Yes	No
ActiveX control creation	Yes	No
COM object creation	Yes	No
OOP : classes, interfaces, properties	Yes	No
PictureBox containers	Yes	No
Desktop Windows ActiveX controls	Yes	No
User-defined types	Yes	No

Setiap fail yang dibina menggunakan EVB akan diberikan nama **.ebp** untuk projek, **.ebf** untuk Form dan **.frx** untuk binari.

ii. Microsoft eMbedded Visual C++ 3.0 (EVC)

Perisian yang telah direka dan ditulis semula untuk pembangunan aplikasi Windows CE. Hampir menyerupai VC++ yang biasa dari segi fungsi dan tatacara. Fail ruang kerja untuk EVC dinamakan **.vcw**, **.vcp** untuk projek EVC biasa dan untuk program normal menggunakan **.exe**.

iii. Microsoft Windows® Platform SDK for Pocket PC

Perisian ini merupakan simulasi sistem pengendalian Windows CE 3.0 untuk pembangunan aplikasi dan permainan beranimasi. Ia hampir menyerupai Pocket PC yang sebenar segi fungsi, antaramuka dan lain-lain termasuk *tools*, *headers*, *lib*, contoh kod dan dokumentasi. Ia merupakan satu pakej

pembangunan teknologi dan produk untuk Pocket PC. Di bawah merupakan spesifikasi system yang diperlukan untuk pembangunan Pocket PC :

- Pemproses : Pentium III-600 / Athlon-600
- Memori : 128MB PC-100 SDRAM
- Hard Drive : 20GB (5400 rpm)
- Kad Video : 8MB AGP 2x
- Monitor : 19 inci

3.5 PEMILIHAN PERKAKASAN

Untuk memastikan permainan beranimasi untuk Pocket PC ini berjaya dibangunkan, beberapa perkakasan yang diperlukan dikenalpasti. Perkakasan ini merupakan persekitaran pembangunan sistem dan pilihannya adalah berkaitan dengan perisian dan kaedah yang digunakan. Di antara perkakasan yang akan dipilih ialah :

- Komputer peribadi dengan pemproses *Intel Pentium III 450 MHz*
 - untuk membangunkan permainan beranimasi dan sebagai pelantar untuk semua perisian yang dipilih.
- Papan kekunci dan tetikus
 - perkakasan input untuk menulis aturcara dan peralatan grafik.
- Monitor
 - alatan antaramuka komputer dengan pengguna.
- Storan cakera keras 30 GB
 - untuk menyimpan(*save*) projek yang dibangunkan.
- 1.44 cakera *floppy*
 - peralatan menyimpan dokumentasi projek.

- 8 x 4 x 32 *Creative CDRW*
 - membuat salinan projek ke cakera padat.
- Pencetak (*printer*)
 - untuk mencetak dokumentasi.

Kesemua perkakasa ini hendaklah dipastikan dalam keadaan baik dan dapat digunakan sepanjang projek ini dijalankan.

4.1 PENGENALAN REKABENTUK

Rekabentuk sistem merupakan salah satu fasa yang mesti ada dalam sesebuah projek. Setiap keperluan dan maklumat yang telah dikumpulkan diubah ke bentuk model dan perincian yang boleh diukur kualitinya sebelum proses pengkodan dijalankan. Dalam fasa ini, rekabentuk antara muka atau skrin sistem dihasilkan sebagai penekanan yang boleh dilihat oleh pengguna dan rekabentuk pengguna.

Rekabentuk antara muka bermula dengan teks amaran model sistem oleh pengguna. Sebagai contoh, pengguna akan melihat skrin awal yang menunjukkan pengiraan. Sebagai contoh, pengguna akan melihat skrin awal yang menunjukkan pengiraan.

BAB 4 REKABENTUK

- Model sistem yang akan dihasilkan oleh pengguna akan

- Model sistem yang akan dihasilkan oleh pengguna akan

- Model sistem yang akan dihasilkan oleh pengguna akan

berikut adalah contoh dan konsep.

Sebagai contoh, perancang Berorientasi Untuk Pocket PC ini, terdapat 2 jenis maklumat yang telah dihasilkan.

1. rekabentuk sistem
2. rekabentuk pengguna

4.0 PENGENALAN REKABENTUK

Rekabentuk sistem merupakan salah satu fasa yang mesti ada dalam sesebuah projek. Setiap keperluan dan maklumat yang telah dikumpulkan diubah ke bentuk model dan perwakilan yang boleh dinilai kualitinya sebelum proses pengkodan dijalankan. Dalam fasa ini, rekabentuk antaramuka atau skrin sistem dihasilkan sebagai penekanan yang boleh menarik minat dan kehendak pengguna.

Rekabentuk antaramuka perlu dinilai dengan teliti supaya mudah difahami oleh pengguna. Selain itu, struktur sistem juga diberikan perhatian supaya mudah dikendalikan pengguna. Semua ini adalah gambaran awal bagaimana sistem tersebut berfungsi.

Biasanya, rekabentuk sistem dimulakan dengan rekabentuk antaramuka diikuti dengan rekabentuk pangkalan data. Rekabentuk yang baik hendaklah sekurang-kurangnya mempunyai ciri-ciri yang berikut :

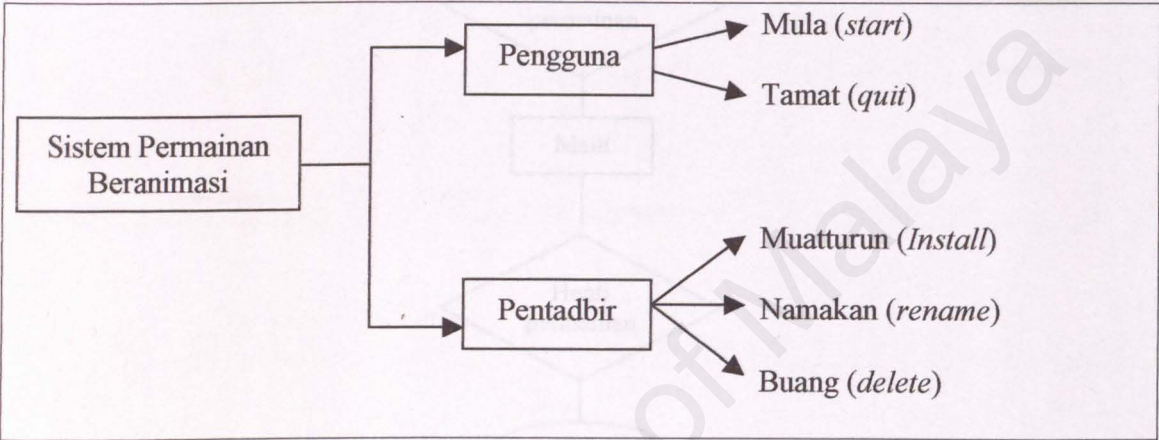
- Sesuai – pemilihan warna, kandungan, saiz, animasi, grafik dan persembahan keseluruhan.
- Mudah – senang difahami dan dikendalikan oleh pengguna atau *user-friendly*.
- Jelas – mestilah dapat memberikan penerangan ringkas yang pengguna faham.
- Konsisten – untuk mengelakkan kekeliruan berlaku, sistem haruslah seimbang dan konsisten.

Bagi projek permainan beranimasi untuk Pocket PC ini, terdapat 2 jenis rekabentuk yang telah dikenalpasti :

1. rekabentuk sistem
2. rekabentuk antaramuka

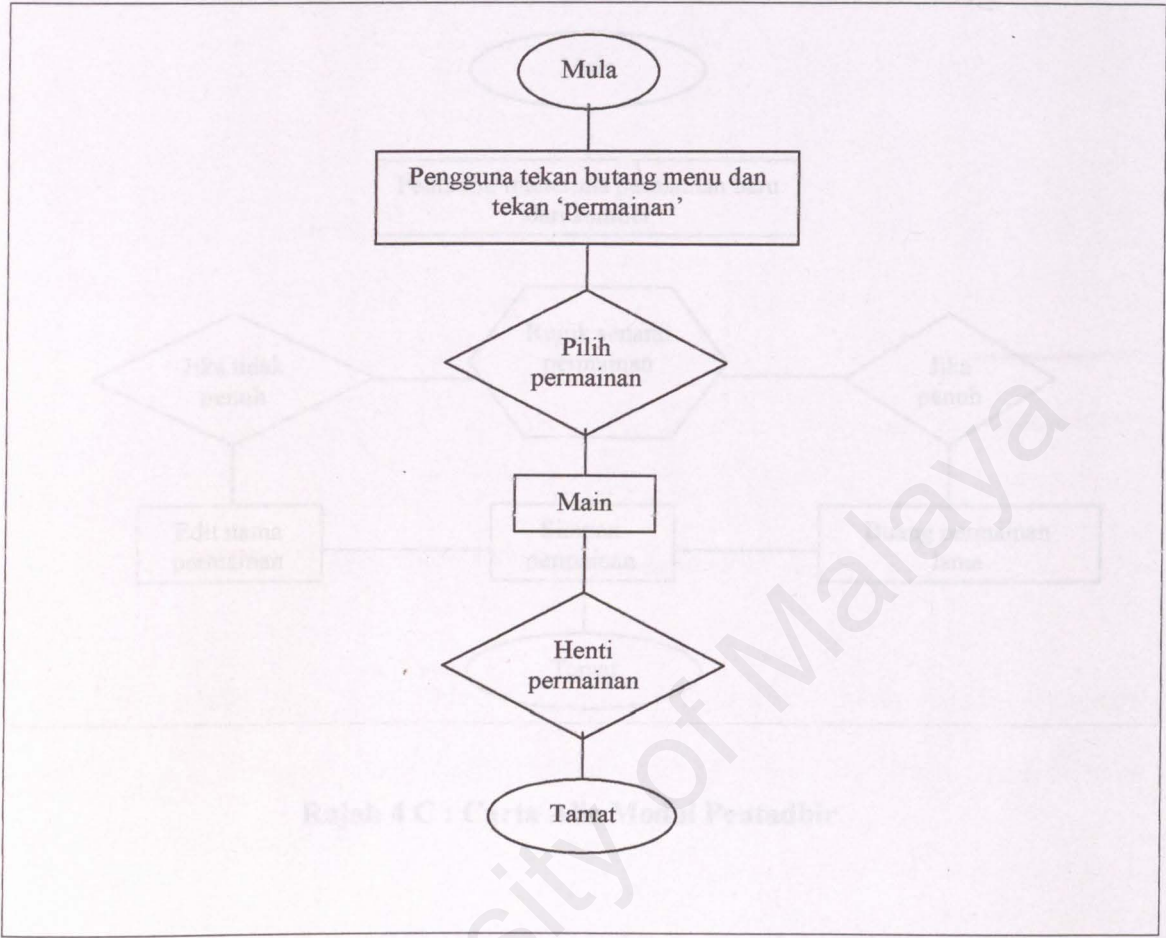
4.1 REKABENTUK SELURUH PERMAINAN BERANIMASI

Rekebentuk sistem boleh ditunjukkan dengan menggunakan carta alir. Carta alir ini menunjukkan hubungan antara modul-modul yang terlibat. Ia menggambarkan struktur pilihan yang terdapat dala sesuatu menu. Rajah di bawah menunjukkan rekabentuk keseluruhan sistem permainan ini :



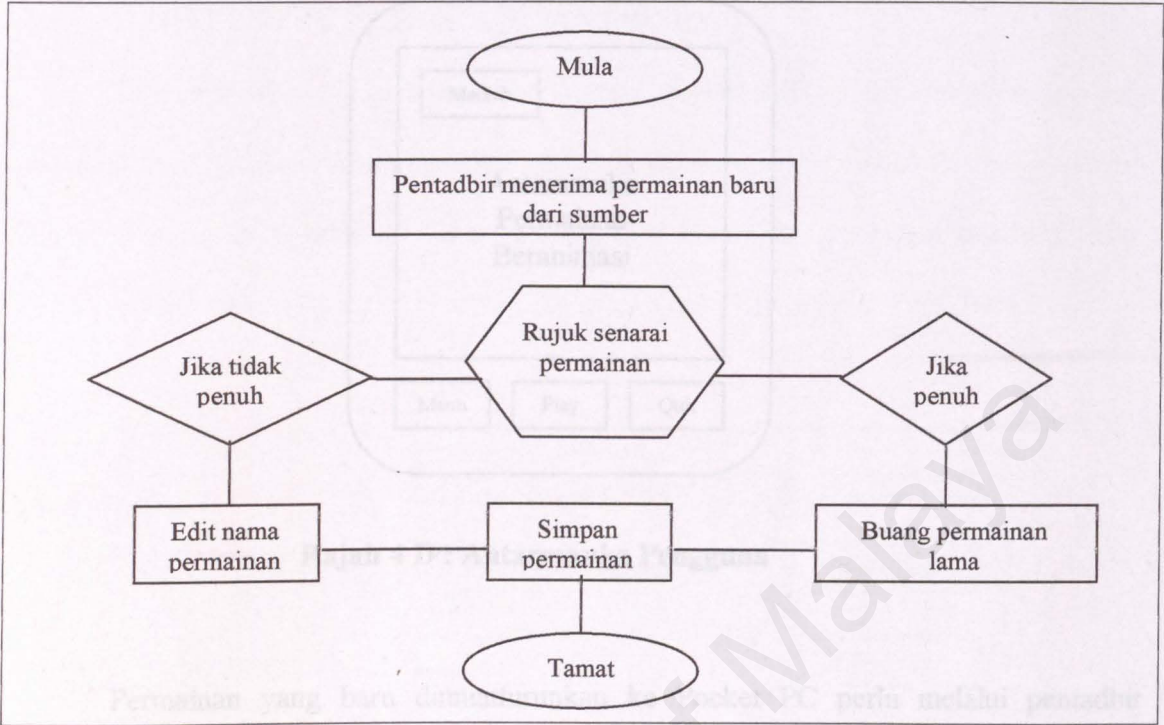
Rajah 4 A : Carta Alir Keseluruhan Sistem

Pengguna hendaklah mengikuti prosedur yang telah ditetapkan untuk memainkan permainan ini. Rajah di bawah boleh memberikan panduan kepada pengguna bagaimana hendak bermain permainan Pocket PC :



Rajah 4 B : Carta Alir Modul Pengguna

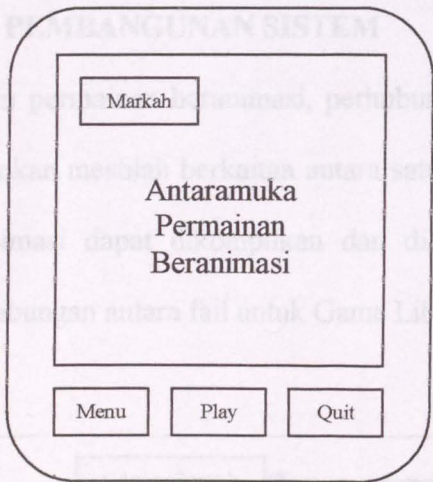
Bagi modul pentadbir pula prosedur tertentu telah ditetapkan untuk menyimpan dan mengedit permainan baru di dalam Pocket PC. Rajah di bawah menunjukkan bagaimana prosedur tersebut berlaku :



Rajah 4 C : Carta Alir Modul Pentadbir

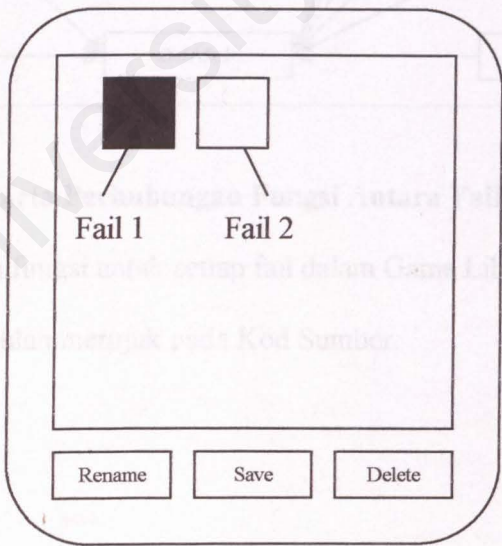
4.2 REKABENTUK ANTARAMUKA PERMAINAN

Antaramuka permainan adalah sangat penting memandangkan di sinilah kebanyakan interaksi di antara pengguna dan sistem berlaku. Namun begitu, antaramuka Pocket PC tidaklah sekompleks antaramuka komputer seperti Windows 98 atau Windows 2000. Antaramuka Pocket PC adalah ringkas tapi padat dan mudah difahami kerana tidak banyak grafik dan paparan disediakan. Rajah di bawah adalah gambaran antaramuka permainan yang akan dibangunkan di dalam Pocket PC :



Rajah 4 D : Antaramuka Pengguna

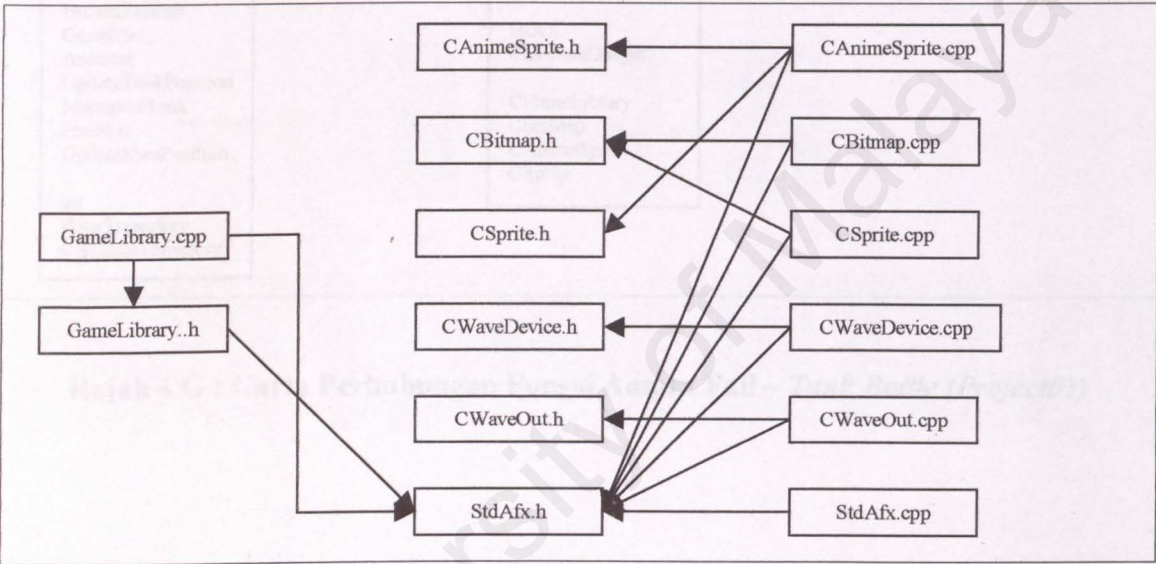
Permainan yang baru dimuatturunkan ke Pocket PC perlu melalui pentadbir memori untuk disimpan atau diedit terlebih dahulu. Rajah di bawah menunjukkan antaramuka bagi pentadbir :



Rajah 4 E : Antaramuka Pentadbir

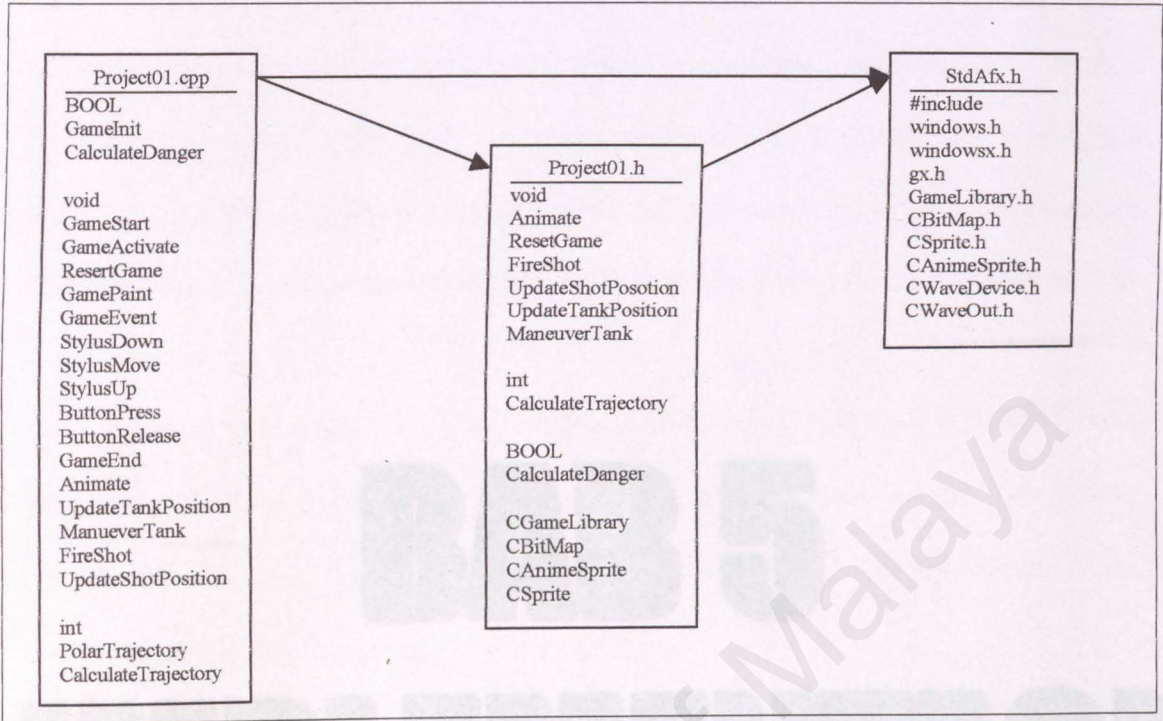
4.3 REKABENTUK PEMBANGUNAN SISTEM

Bagi pembangunan permainan beranimasi, perhubungan setiap fungsi antara fail-fail yang hendak dibangunkan mestilah berkaitan antara satu sama lain. Tujuannya adalah supaya permainan beranimasi dapat dikompilkan dan dilarikan tanpa sebarang ralat. Berikut adalah carta perhubungan antara fail untuk Game Library dan Tank Battle :



Rajah 4 F : Carta Perhubungan Fungsi Antara Fail – *Game Library*

* Oleh kerana panggilan fungsi untuk setiap fail dalam Game Library panjang dan ruang tidak mencukupi, bolehlah merujuk pada Kod Sumber.



Rajah 4 G : Carta Perhubungan Fungsi Antara Fail – *Tank Battle (Project01)*

3.4 PENGENALAN IMPLEMENTASI SISTEM

Untuk mengembangkan permainan beranimasi ini untuk Pocket PC ini, pelbagai proses yang berkaitan dijalankan sepanjang pembangunan. Proses implementasi ini merupakan proses yang mengenalpasti keadaan dan kedudukan sistem yang sebenar. Dengan kata lain, sistem diimplementasikan untuk memuktamadkan sesuatu yang berkaitan yang berbentuk abstrak ke bentuk objek yang boleh digunakan (kod-kod program).

Tujuan dari pengembangan sistem ialah untuk menghasilkan kod program yang jelas, mudah dan berkesan. Selain itu, sistem yang dihasilkan juga haruslah mempunyai ciri-ciri yang baik. Dengan kata lain, sistem yang dihasilkan, dapat memberi pelbagai manfaat dan kegunaan kepada pengguna.

BAB 5 IMPLEMENTASI

Pengembangan sistem melibatkan pelbagai proses yang berkaitan dengan pembangunan sistem. Proses pembangunan sistem melibatkan pelbagai proses yang berkaitan dengan pembangunan sistem. Proses pembangunan sistem melibatkan pelbagai proses yang berkaitan dengan pembangunan sistem.

5.1.1 Spesifikasi Perisian

Untuk menjalankan projek permainan beranimasi untuk Pocket PC ini, beberapa perkakasan telah diidentifikasi dalam proses pembangunan projek. Berikut adalah senarai perkakasan yang akan digunakan.

- **Prosesor** - Pentium III 450MHz

- **RAM** - 256MB SDRAM

- **Cikera keras** - 20GB

5.0 PENGENALAN IMPLEMENTASI SISTEM

Untuk membangunkan permainan beranimasi ini untuk Pocket PC ini, pelbagai proses yang berkaitan dijalankan sepanjang pembangunan. Proses implementasi ini merupakan proses yang mengenalpasti keadaan dan kedudukan sistem yang sebenar. Dengan kata lain, sistem diimplementasikan untuk menukarkan fasa rekabentuk yang berbentuk abstrak ke bentuk objek yang boleh digunakan (kod-kod program).

Tujuan fasa implementasi sistem ialah untuk menghasilkan kod sumber yang jelas, mudah dan beserta dokumentasi yang memudahkan proses pengujian, pengubahsuaian dan penyelenggaraan. Semasa pengkodan sistem dilakukan, dapat dilihat pelbagai masalah dan ralat boleh didapati.

5.1 PERSEKITARAN PEMBANGUNAN

Persekitaran pembangunan memainkan peranan penting dalam kitar hayat pembangunan sistem permainan. Pemilihan perkakasan dan perisian pembangunan yang tepat dapat mempercepatkan proses pembangunan dan menentukan berjaya atau gagal projek tersebut.

5.1.1 Keperluan Perkakasan

Untuk menjayakan projek permainan beranimasi untuk Pocket PC ini, beberapa perkakasan telah dikenalpasti dalam proses pembangunan projek. Berikut adalah senarai perkakasan yang akan digunakan.:

- Prosesor – Pentium III 450MHz
- 256MB SDRAM
- Cakera keras – 30GB

- Peranti input – papan kekunci dan tetikus
- Peranti output – monitor dan pencetak

5.2 Keperluan Perisian

Di bawah merupakan penerangan ringkas mengenai perisian yang digunakan untuk pembangunan sistem :

- **Sistem Pengendalian Microsoft Windows 2000**

- sebagai pelantar untuk membangunkan projek menggunakan simulasi yang disediakan. Malahan, untuk membangunkan projek perlu menggunakan system pengendalian ini sahaja kerana beberapa perisian lain tidak menyokong Microsoft Windows 98 SE.

- **Microsoft Windows® Platform SDK for Pocket PC**

- dengan menggunakan sistem pengendalian Windows CE 3.0 (terkini), simulasi ini dikhaskan untuk para pembangun aplikasi dan permainan beranimasi untuk Pocket PC. Perisian ini boleh dimuatturunkan dari www.microsoft.com/mobile dan bersaiz 64.8MB sahaja. Penggunaannya hampir sama dengan Pocket PC yang benar cuma hanya ada beberapa aplikasi yang tidak dapat digunakan dalam simulasi ini.

- **Microsoft eMbedded Visual Basic 3.0 (EVB)**

- merupakan salah satu perisian yang utama dalam membangunkan permainan beranimasi dan aplikasi untuk Pocket PC. Kebanyakan aplikasi asas Windows CE dibangunkan menggunakan perisian ini. Perisian ini digunakan untuk membangunkan permainan beranimasi yang bercirikan kepada antaramuka pengguna. Contoh di bawah adalah beberapa kod yang digunakan dalam EVB :


```

Option Explicit
Dim mode, total
Private Sub Form_Load ( )
    Randomize
    Mode = 1
    Total = 0
    Timer.Interval1 = 10
    Timer.Interval2 = 1000
End Sub

```

- **Microsoft eMbedded Visual C++ 3.0 (EVC)**

- menyerupai Visual C++ yang biasa tetapi khas untuk Pocket PC.

Kebanyakan permainan beranimasi dibina menggunakan perisian ini memandangkan ciri-ciri permainan perlu dibangunkan menggunakan pengaturcaraan. Mempunyai fungsi-fungsi yang menyokong beberapa pemproses simulasi Pocket PC Win 32 (ARM, MIPS, SH3 dan x86em). Untuk membina program EVC, terlebih dahulu memilih apakah pelantar yang akan digunakan untuk melarikan program. Contoh di bawah adalah beberapa contoh kod yang wujud dalam EVC :

```

#define WIN32_LEAN_AND_MEAN
#include "stdafx.h"
#include <windows.h>
#include <windef.h>
#include "gx.h"
#define GAME_TIMER 101

```

```

GXKeyList gxKeys;
TCHAR szButton[20];
ATOM MyRegisterClass (HINSTANCE hInstance, LPTSTR
    szWindowsClass);
BOOL InitInstance (HINSTANCE hInstance, int CmdShow);
LRESULT CALLBACK WndProc (HWND hWnd, UINT message,
    WPARAM wParam, LPARAM lParam);
void CALLBACK TimerProc (HWND hwnd, UINT uMsg, UINT
    idEvent, DWORD dwTime

```

5.2 PEMBANGUNAN SISTEM

Permainan beranimasi ini akan dibangun menggunakan model prototaip. Oleh itu, satu set kod program ditulis terlebih dahulu dengan berasaskan maklumat yang telah dicarai terlebih dahulu. Penulisan kod dibuat secara berperingkat-peringkat dan hendaklah kesemua kod ini disambungkan antara satu sama lain supaya ia dapat dilarikan.

Sebelum pengkodan dilakukan, perisian yang telah dimuatturunkan perlu melakukan beberapa perkara terlebih dahulu. Di bawah adalah perkara-perkara yang perlu dibuat :

- ✓ muatturunkan fail GameAPI[6] dari laman web Microsoft
- ✓ salin fail **gx.lib** dan **gx.dll** ke direktori “\windows” dalam simulator Pocket PC supaya program dapat dilarikan tanpa masalah
- ✓ salin fail **gx.h** ke direktori “wce300\MS Pocket PC\include”. Tujuan agar pengkompil program dapat menerima arahan “**#include <gx.h>**”.

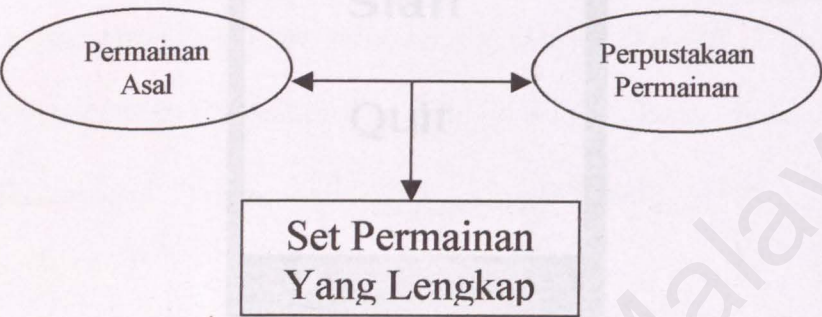
Secara keseluruhan, sebanyak 3 permainan beranimasi akan dilarikan di dalam projek ini iaitu :

- a. Bebola anjalan (*blackout*) – yang dibangun menggunakan EVB
- b. Bebola anjalan (*blackout*) – yang dibangun menggunakan EVC
- c. Kereta kebal – pengaturcaraannya lebih kompleks

5.2.1 Peringkat Pembangunan

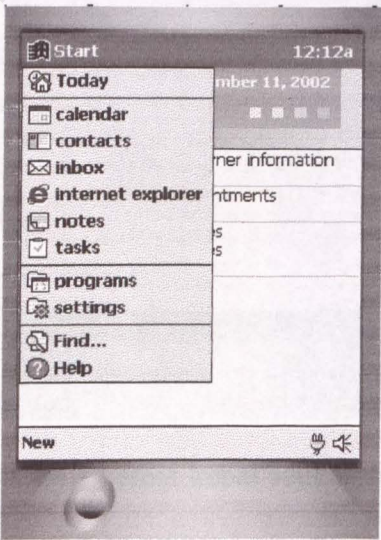
Untuk membangunkan permainan beranimasi ini, beberapa peringkat pengkodan perlu dilakukan kerana ia merupakan keperluan dalam menjalankan program dengan sempurna. Peringkat-peringkat ini ialah :

- ✓ Membina kod permainan yang hendak dijalankan secara asas
- ✓ Membina kepelbagaian perpustakaan permainan (*game library*) untuk Pocket PC
- ✓ Mengubahsuai permainan asal agar memenuhi keperluan perpustakaan permainan untuk kelancaran program

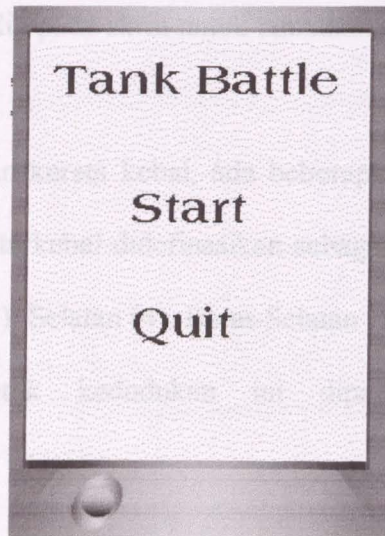


Rajah 5-A : Hubungan lengkap dalam membangunkan program

Untuk permainan beranimasi ini berfungsi dengan baik setelah siap, beberapa prosedur perlu difahami untuk menjalankannya. Berikut beberapa langkah yang dilakukan untuk menjalankan permainan :

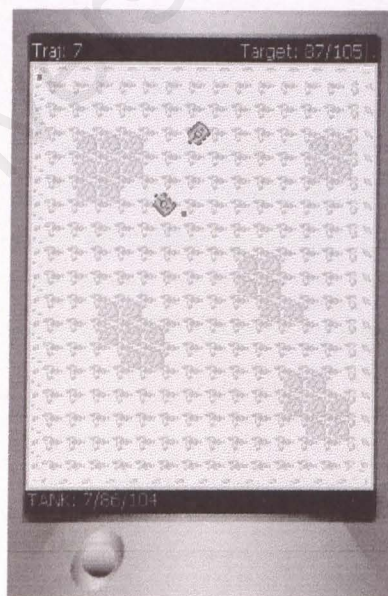


1. Mengikut urutan, klik pada menu *Start, Program, Games, Tank Battle* dan *Project01*. Fail *Project01* ini akan memulakan permainan.



2. Paparan di atas menunjukkan tajuk permainan, mula dan tamatkan permainan.

Untuk mula bermain, klik pada *Start*. Jika tidak mahu bermain, klik sahaja pada *Quit*.



3. Dapat dilihat di sini terdapat 2 kereta kebal sedang berperang dengan menembak antara satu sama. Kereta kebal digerakkan dengan menggunakan kekunci atas, kiri,

kanan dan bawah. Untuk menembak pula tekan kekunci *Space*. Kereta kebal yang dikawal oleh komputer akan cuba menembak pada kereta kebal pengguna dan boleh mengelak dari peluru. Titik yang berwarna merah adalah peluru yang ditembak. Jika tidak mahu bermain lagi, klik pada skrin untuk tamatkan permainan.

Untuk menggerakkan kereta kebal, ada beberapa cara pengkodan yang perlu dilakukan. Kedudukan kereta kebal didefinisikan sebagai Utara (0), Utara-Timur (1), Timur (2), Timur-Selatan (3), Selatan (4), Barat-Selatan (5), Barat (6) dan Utara-Barat (7). Nombor-nombor untuk kedudukan ini dipanggil dalam fungsi void *ManeuverTank*.

```
void GamePaint(HWND hWnd) // untuk warna
void GameEvent()
void StylusDown(int x, int y)
void StylusMove(int x, int y)
void StylusUp(int x, int y)
void ButtonPress(int hButtonID, POINT pt)
void ButtonRelease(int hButtonID, POINT pt)
void GameStart() // untuk program
void Animate(HWND hWnd) // untuk animasi
void UpdateTankPosition(int TankID, int x, int y) // untuk kereta kebal
int PolarTrajectory(int TankID, int x, int y) // untuk arah kereta kebal
void ManeuverTank(int TankID, int hButtonID) // menggerakkan tank
void CalculateTrajectory(int TankID, POINT pShot, int hButtonID) // untuk arah kereta kebal
void FireShot(int TankID) // menembak kereta kebal
void UpdateShotPosition(int TankID, int x, int y) // untuk arah kereta kebal
```

5.3 PENGKODAN UTAMA

Pengaturcaraan untuk permainan beranimasi adalah sama sekali berlainan dengan pengaturcaraan yang telah dipelajari di kelas. Malah, ia juga berlainan dengan pengkodan untuk sebarang aplikasi. Di sebelah merupakan beberapa panggilan fungsi yang utama untuk membangunkan pengkodan permainan beranimasi untuk Pocket PC.

5.3.1 Pengkodan Permainan Beranimasi (*Project01*)

```

BOOL GameInit(HINSTANCE hInst) // sebelum masuk windows
void GameStart(HWND hWnd) // selepas masuk windows
void GameActivate(HWND hWnd) // selepas permainan bermula
void ResetGame() // kenalpasti semua variables
void GamePaint(HWND hWnd) // untuk warna
void GameEvent()
void StylusDown(int x, int y)
void StylusMove(int x, int y)
void StylusUp(int x, int y)
void ButtonPress(int iButtonID, POINT pt)
void ButtonRelease(int iButtonID, POINT pt)
void GameEnd() // akhiri program
void Animate(HDC hdc) // rutin utama animasi
void UpdateTankPosition(int iTank) // kenalpasti posisi X, Y kereta kebal
int PolarTrajectory(int iTraj) // kira kemungkinan posisi musuh
void ManeuverTank(int iTank, int iEnemy) // menggerakkan tank
int CalculateTrajectory(POINT A, POINT B) // pastikan arah pada 2 titik
BOOL CalculateDanger(int iTank, POINT pShot, int iTraj) // kira bahaya
void FireShot(int iTank) // menembak peluru
void UpdateShotPosition(int iShot) // kenalpasti posisi X, Y peluru

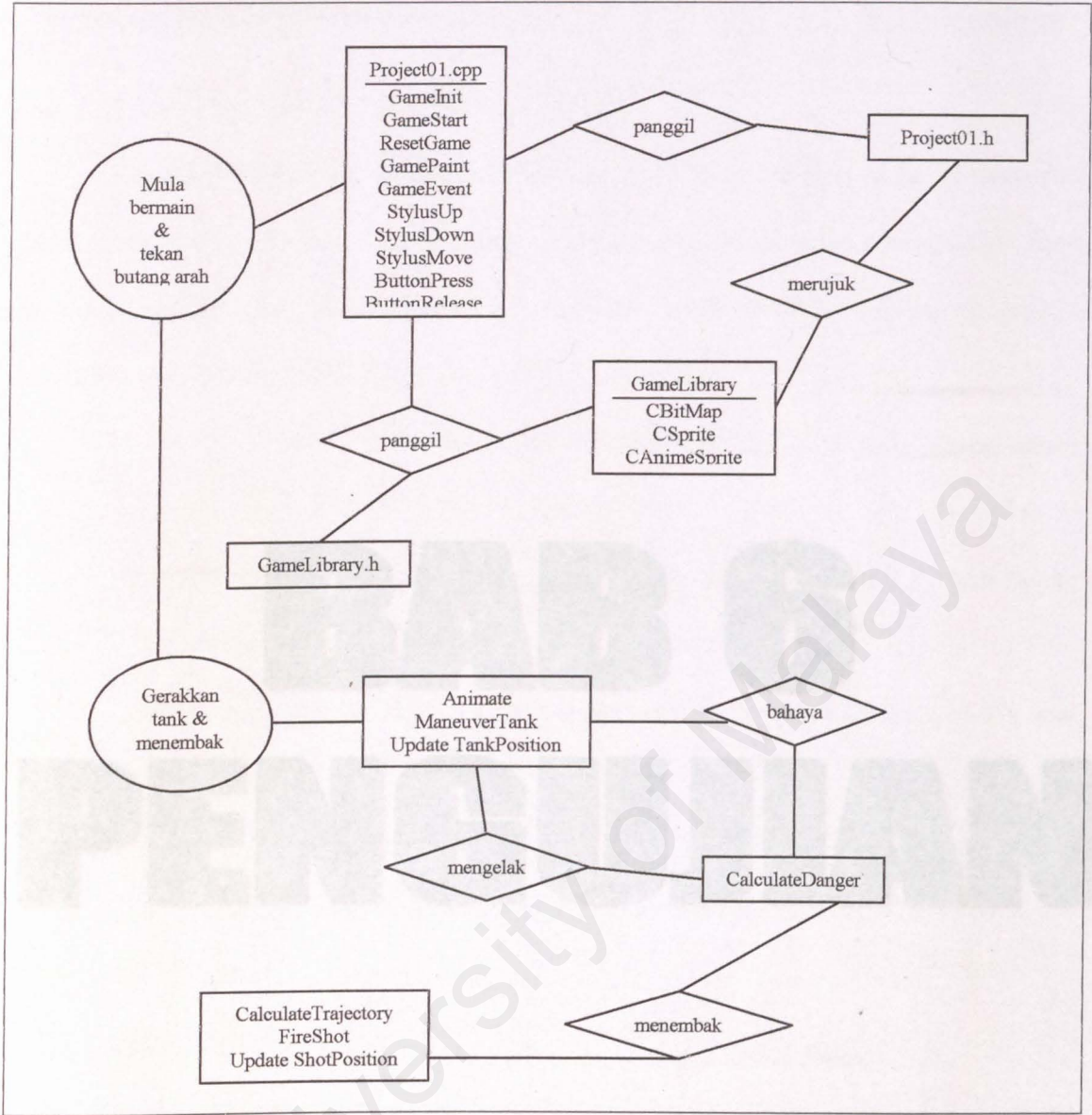
```


5.3.2 Pengkodan GameLibrary

```

CGameLibrary::CGameLibrary(HINSTANCE hInst, LPTSTR szNewWindowClass)
CGameLibrary::~CGameLibrary() //constructor dan destructor
BOOL CGameLibrary::Initialize(int nCmdShow)
LRESULT CGameLibrary::EventHandler(HWND hWnd, UINT message,
                                WPARAM wParam, LPARAM lParam)
void CGameLibrary::Error(LPTSTR szError) //kawal mesej ralat
void CGameLibrary::Shutdown() //hantar mesej tutup windows pada akhir program
void CGameLibrary::G_BeginDraw()
void CGameLibrary::G_EndDraw()
BOOL CGameLibrary::G_ClearScreen(COLOR color)
int CGameLibrary::G_DrawPixel16(unsigned char *VidMem, int X, int Y, COLOR color)
void CGameLibrary::G_DrawSolidRect(unsigned char *VidMem, int iLeft, int iTop,
                                int iRight, int iBottom, COLOR color)
void CGameLibrary::FatalError(LPTSTR lpStr)
void CGameLibrary::PrintText(HDC hdc, LPCTSTR strText, int row, int col,
                            COLORREF color, int iBkMode)
void CGameLibrary::PrintTextXY(HDC hdc, LPCTSTR strText, int x, int y,
                            COLORREF color, int iBkMode)
LPWSTR CGameLibrary::GetPath(TCHAR *filename)
void CGameLibrary::Delay(int ms)
BOOL CGameLibrary::PointInside(POINT pt, int left, int top, int right, int bottom)
BOOL CGameLibrary::PointInside(POINT pt, RECT rt)
LRESULT CALLBACK WndProc(HWND hWnd, UINT message,
                            WPARAM wParam, LPARAM lParam)
int WINAPI WinMain( HINSTANCE hInstance, HINSTANCE hPrevInstance,
                  LPTSTR lpCmdLine, int nCmdShow )
void MsgBox(LPTSTR szMessage)

```



Rajah 5 B : Mulakan Permainan, Menggerak dan Menembak

6.1. PENGENALAN PENGUJIAN SISTEM

Pengujian sistem merupakan proses yang dilakukan setelah kerja pengkodean untuk mengungkap titik dan masalah dalam pelaksanaan sistem yang dibangun. Suatu cara atau pendekatan yang teratur dan berstruktur perlu digunakan dalam melakukan pengujian untuk memastikan kualitas produk yang dihasilkan.

Untuk mencapai tujuan pengujian, beberapa parameter tertentu perlu diikuti

BAB 6

PENGUJIAN

6.1. STRATEGI PENGUJIAN

Terdapat beberapa strategi pengujian yang dapat digunakan dalam mengorganisir pengujian, antara lain:

1. Pengujian integrasi
2. Pengujian sistem
3. Pengujian penerimaan

6.0 PENGENALAN PENGUJIAN SISTEM

Pengujian sistem merupakan proses yang dilakukan selepas kerja pengkodan untuk mengesan ralat dan masalah dalam pelaksanaan sistem yang dibangunkan. Satu cara atau pendekatan yang teratur dan berstruktur perlu digunakan dalam melakukan pengujian untuk menentukan kualiti perisian yang dihasilkan.

Untuk memenuhi objektif pengujian, beberapa peraturan tertentu perlu diikuti iaitu :

- Pengujian merupakan suatu proses pelaksanaan program dengan tujuan untuk mencari ralat dan jalan penyelesaiannya
- Kes pengujian yang baik adalah kes yang mempunyai kebarangkalian ralat yang tinggi
- Pengujian yang berjaya ialah yang dapat mengenalpasti dan mendedahkan ralat yang tidak dapat didedahkan kemudiannya

6.1 STRATEGI PENGUJIAN

Terdapat beberapa strategi pengujian yang dapat digunakan dalam membangunkan permainan beranimasi ini :

- Pengujian unit
- Pengujian integrasi
- Pengujian sistem
- Pengujian penerimaan

6.1.1 Pengujian Unit

Pengujian dilakukan terhadap unit perisian kecil yang dinamakan modul. Ia terdiri daripada satu atau lebih fungsi atau prosedur. Beberapa aspek perlu diambil kira dalam pengujian modul seperti pengendali ralat, antaramuka, nilai sempadan dan kelas data, laluan logik dan struktur data.

Pengujian unit dijalankan bagi memastikan setiap fungsi sistem dilaksanakan dengan betul dan berintegrasi antara satu sama lain. Ada beberapa ujian yang terlibat iaitu :

- **Pengujian Kod** – dilakukan melalui pembacaan dan pengamatan semula kod yang telah ditulis bagi mengesan ralat. Setiap variasi, definisi dan deklarasasi kod dipastikan memenuhi keperluan perisian yang hendak dijalankan.
- **Larian Kod** – kod yang telah siap ditulis dan dikaji dikompilkan dan setiap ralat yang dipaparkan diambil perhatian.

6.1.2 Pengujian Integrasi

Untuk menguji semua modul sekaligus adalah sukar memandangkan semua kod perlu dikompilkan dan dilarikan secara berperingkat. Pengujian perlu dilakukan terhadap semua modul secara integrasi mengikut struktur hirarki perisian untuk mengesan ralat dalam suatu komponen sistem. Kejayaan terhadap pengujian unit tidak bermakna ia tidak memberi masalah kepada pengujian integrasi kerana mungkin ralat wujud ketika cuba dijalankan ke sistem sebenar.

6.1.2 Terdapat dua kaedah untuk menjalankan pengujian integrasi terhadap sistem ini iaitu :

- **Pengujian atas-bawah**

Proses pengujian di mana modul-modul diuji dan digabungkan secara menurun mengikut struktur hirarki perisian. Kesilapan dari fasa rekabentuk dapat dikesan dan dapat mengurangkan kos untuk merekabentuk semula sistem.

- **Pengujian bawah-atas**

Proses pengujian yang dimulakan dari peringkat bawah hirarki perisian. Dengan cara ini, keperluan terhadap ujian-ujian secara berasingan akan berkurangan.

6.1.3 Pengujian Sistem

Tujuannya ialah untuk memastikan semua komponen system dapat digabungkan menjadi satu sistem keseluruhan. Ia boleh dipanggil sebagai pengesanan ralat yang mungkin dari segi interaksi komponen system dengan komponen lain termasuk perkakasan. Selain itu, ia akan mengesahkan samada sistem dapat beroperasi seperti yang dikehendaki. Pengujian ini juga menguji proses baik-pulih dapat dilakukan segera jika ada ralat atau menguji kawalan keselamatan boleh dipercayai sepenuhnya.

6.1.4 Pengujian Penerimaan

Merupakan pengujian yang terakhir sebelum sistem benar-benar lengkap.

Di sini, sistem akan dicuba oleh beberapa pengguna dari pelbagai peringkat menggunakan data sebenar untuk mencapai kehendak dan objektif sebenar.

6.2 JENIS RALAT

Suatu pengujian yang baik boleh mengenal pasti ralat-ralat yang tidak dikesan semasa fasa analisis, rekabentuk dan implementasi. Terdapat 3 jenis ralat yang biasa ditemui :

a) Ralat kompilasi

- terjadi akibat pembinaan kod yang salah disebabkan kesilapan menaip kata kunci atau tertinggal tanda-tanda penting. Biasanya, ralat sintaks dapat dikesan di sini kerana ia akan dipaparkan semasa kompilasi

b) Ralat larian masa

- berlaku ketika pelaksanaan sistem atau semasa sistem dijalankan di mana pernyataan melakukan operasi yang tidak munasabah (cth : pembahagian dengan nilai sifar)

c) Ralat logical

- terjadi apabila aplikasi tidak menghasilkan seperti yang dikehendaki (kesilapan output). Aplikasi mempunyai kod sah secara sintaks, dilarikan tanpa menjalankan operasi yang tidak sah dan menghasilkan jawapan yang salah. Verifikasi dapat dicapai hanya dengan menguji aplikasi dan

menganalisa hasil yang diperolehi. Pengesanan ralat boleh dilakukan oleh pengguna dan pembangun

6.3 PENILAIAN PENGGUNA

Di antara pengujian di peringkat terakhir ialah maklumbalas dari pengguna. Ini merupakan salah satu panduan untuk meningkatkan lagi kualiti sistem untuk memenuhi kepuasan pengguna. Penilaian ini dijalankan setelah semua ralat sistem diperbetulkan dan system dapat berfungsi dengan sempurna.

Dalam menjalankan pengujian pengguna ini, beberapa orang pengguna (gamers) dari pelbagai umur telah dipilih untuk mencuba permainan beranimasi ini. Segala komen, nasihat dan cadangan yang dikemukakan diambil perhatian dalam mengubahsui permainan untuk menjadikan ia lebih baik.

Berikut merupakan rumusan secara ringkas pemain-pemain permainan beranimasi terhadap sistem ini :

- Untuk tahap sistem yang menggunakan Pocket PC, permainan ini menarik dan cukup ringkas. Mudah difahami dan mudah untuk bermain.
- Sangat mudah untuk dimuatturunkan ke Pocket PC kerana tidak memerlukan aplikasi muatturun yang kompleks
- Jika boleh, bina permainan beranimasi yang lebih kompleks dan menarik. Tambah lagi animasi agar pengguna tidak cepat merasa bosan. Gunakan sebarapa banyak aplikasi yang boleh
- Permainan ini boleh dimainkan tak kira tua atau muda dan di mana sahaja asalkan mempunyai Pocket PC.

6.4 GUNA-SEMULA

Untuk pengujian guna-semula, permainan beranimasi ini diuji menggunakan 2 kaedah iaitu kotak-hitam (*black-box reuse*) dan kotak-putih (*white/clear-box reuse*).

6.4.1 Pengujian Kotak-Hitam (*Black-box*)

Permainan beranimasi ini boleh dimainkan tanpa perlu mengubah apa-apa komponen atau pengkodan. Tetapi, ia tidak boleh dimainkan pada pelantar yang berlainan dari Windows CE kerana pengkodan dan jenis fail adalah amat berlainan.

6.4.2 Pengujian Kotak-Putih (*White/Clear-box*)

Untuk mengubah permainan beranimasi ini mengikut citarasa pengguna atau individu, permainan ini boleh diubah komponen atau kodnya supaya permainan menjadi lebih menarik. Contohnya seperti mengubah latar belakang, mengubah bentuk bebola, mengaturcarakan permainan kepada pelbagai kesusahan dan sebagainya. Namun, permainan ini tetap tidak boleh dimainkan pada pelantar selain Windows CE tetapi pengkodannya boleh diambil dan diubah agar dapat diaplikasikan ke Visual Basic atau C++ yang biasa. Ini bermakna satu permainan yang sama boleh dihasilkan dalam pelbagai versi.

7.0 PENYELENGGARAAN SISTEM

Penyelenggaraan adalah fase di mana memastikan peralatan bernomor seragam berada dalam keadaan yang baik dan terpelihar. Taktik penyelenggaraan ialah proses pengalihan sistem atau model selepas digil bertujuan membetulkan kecacatan untuk meningkatkan kegunaan dan kualiti sistem dan menyesuaikan dengan pertumbuhan.

Pengalihan model perlu dilakukan di setiap fasa pembinaan sistem supaya sistem dapat mengikuti setiap perubahan yang dilakukan pada model sedia ada. Anggaran kos untuk pengalihan model akan berbeza kepada 4 aspek berikut:

BAB 7

PENYELENGGARAAN

7.1 JENIS-JENIS PENYELENGGARAAN

Penyelenggaraan boleh diklasifikasikan kepada beberapa kategori iaitu:

- 1) Penyelenggaraan pembaikan
- 2) Penyelenggaraan pemeliharaan
- 3) Penyelenggaraan penyempurnaan
- 4) Penyelenggaraan pencegahan

7.0 PENYELENGGARAAN SISTEM

Penyelenggaraan adalah fasa di mana memastikan permainan beranimasi sentiasa berada dalam keadaan yang baik dan tanpa ralat. Takrif penyelenggaraan ialah proses pengubahsuaian sistem atau modul selepas diuji bertujuan membetulkan kesilapan untuk meningkatkan keupayaan dan kualiti sistem dan menyesuaikan dengan persekitaran.

Penyelenggaraan perlu dilakukan di setiap fasa pembinaan sistem supaya sistem dapat menampung segala perubahan yang dilakukan pada modul-modul atau fungsi-fungsi. Fokus penyelenggaraan kepada prestasi sistem bergantung kepada 4 aspek berikut :

- a. Kawalan penyelenggaraan ke atas fungsi sistem dari masa ke semasa
- b. Kawalan penyelenggaraan ke atas modifikasi sistem iaitu sebarang perubahan yang dilakukan ke atas sistem perlu diselenggarakan
- c. Memastikan fungsi yang diterima benar-benar lengkap dan sempurna
- d. Mengekalkan prestasi sistem secara konsisten

7.3 JENIS-JENIS PENYELENGGARAAN

Penyelenggaraan boleh dikelaskan kepada beberapa kategori iaitu :

- 1) Penyelenggaraan pembaikan
- 2) Penyelenggaraan penyesuaian
- 3) Penyelenggaraan penyempurnaan
- 4) Penyelenggaraan pencegahan

sistem yang sedang dibangun. Terdapat 2 jenis dokumentasi yang boleh menjadi panduan utama kepada permainan beranimasi ini iaitu :

1) Manual pengguna

Satu dokumentasi bertulis yang lengkap kepada pengguna dan pentadbir dengan edisi yang berlainan :

- a. Edisi pengguna – sebagai rujukan pengguna yang ingin bermain permainan beranimasi tentang cara bermainnya.
- b. Edisi pentadbir – panduan kepada pengguna mahir tentang bagaimana hendak memuat turun permainan ini ke perkakasan atau simulasi Pocket PC.

2) Dokumentasi dalam sistem

Segala kod sumber pembangunan permainan beranimasi dan komponen-komponennya dipaparkan untuk rujukan pembangun permainan yang lain.

7.3 PENILAIAN SISTEM

7.3.1 Kelebihan system

1) Antaramuka yang ramah pengguna

- antaramuka yang dibangun adalah mudah difahami oleh semua peringkat pengguna memandangkan skrin Pocket PC tidaklah sekompleks komputer peribadi

2) Mutaturun permainan yang mudah

- panduan untuk memuat turun permainan ini diberikan dan prosedurnya cepat dan tidak merumitkan

3) Tidak terlalu banyak input diperlukan

- memandangkan Pocket PC tidak mempunyai papan kekunci, adalah mudah untuk memainkan permainan menggunakan *Stylus* saja

7.3.2 Kekurangan system

- 1) Memerlukan perisian khas untuk memuatturunkan permainan
 - untuk memuatturkan permainan memerlukan perisian yang dinamakan *ActiceSync*[7] kerana ia menyokong rangkaian antara komputer peribadi dan Pocket PC
- 2) Permainan tidak mempunyai bunyi
 - boleh untuk memasukkan bunyi tetapi memerlukan perisian yang khas untuk merakan bunyi untuk Pocket PC yang perlu di ambil dari laman web tertentu. Tambahan, masa pembangunan tidak mencukupi untuk meningkatkan lagi kehebatan permainan beranimasi ini.
- 3) Tidak dapat menggunakan *DirectX*
 - permainan beranimasi menggunakan GAPI untuk menjalankan permainan kerana Pocket PC tidak dapat menampung kuasa yang besar dari *DirectX*. *DirectX* hanya dapat digunakan untuk permainan untuk komputer peribadi.

7.3.3 Harapan Pada Masa Hadapan

Permainan beranimasi untuk Pocket PC merupakan titik permulaan yang paling baik untuk individu belajar membangunkan permainan menggunakan pengaturcaraan. Pembinaanya tidaklah begitu susah, hanya pengkodannya mempunyai perbezaan berbanding pengaturcaraan aplikasi. Malahan, terdapat banyak

laman web yang memberikan perkhidmatan mengajar membina permainan beranimasi secara percuma.

Untuk projek permainan beranimasi untuk Pocket PC ini, masih terdapat banyak lagi kekurangan yang perlu diperbetulkan. Projek ini merupakan projek yang membina permainan yang mudah dan masih jauh lagi untuk membina permainan yang lebih kompleks. Perjalanan permainan boleh dipertingkatkan lagi dengan menambah pilihan (options) supaya permainan lebih mencabar. Antaramukanya boleh ditambah dengan warna yang lebih menarik. Oleh kerana Pocket PC merupakan salah satu alat perhubungan masa kini, permainan ini boleh dikodkan supaya boleh bermain secara rangkaian dengan orang lain atau komputer peribadi. Selain itu, permainan beranimasi boleh dicipta dalam pelbagai genre supaya lebih menarik dan pelbagai pilihan.

Banyak lagi yang perlu dipelajari untuk membina permainan yang lebih susah. Tetapi, bagi negara Malaysia yang masih belum lagi membangun dari segi permainan beranimasi, ini merupakan bahagian terbaik untuk memulakannya. Diharap pada masa akan datang, projek ini dapat menjadi rujukan untuk membina permainan untuk komputer peribadi pula malahan lebih kompleks dan menarik daripada permainan sekarang.

8.0 MASALAH DAN PENYELESAIANNYA

Sebelumnya proyek ini dibayangkan, berbagai masalah telah dihadapi tidak kira dalam segi huraian. Kerana fasa pembangunan permainan beranimasi ini dijalankan, apabila masalah yang timbul akan cuba diselesaikan dengan cara yang terbaik dan pantas agar projek ini dapat dilaksanakan. Berikut merupakan beberapa masalah yang dihadapi ketika pembangunan dijalankan:

1) Ketidaktepatan dalam menggunakan masa

Proyek yang telah selesai ini ialah Permainan Beranimasi Untuk Pocket PC. Dalam projek ini, saya telah menghadapi beberapa masalah ke arah akhir projek ini. Masalah yang dihadapi dalam projek ini adalah berkaitan dengan masa yang digunakan dalam projek ini. Masalah ini timbul kerana saya tidak dapat mengawal masa yang digunakan dalam projek ini.

BAB 8

MASALAH & KESIMPULAN

Salah satu masalah yang dihadapi dalam projek ini adalah berkaitan dengan masa yang digunakan dalam projek ini. Masalah ini timbul kerana saya tidak dapat mengawal masa yang digunakan dalam projek ini. Masalah ini timbul kerana saya tidak dapat mengawal masa yang digunakan dalam projek ini.

2) Ketidaktepatan dalam menggunakan masa

Salah satu masalah yang dihadapi dalam projek ini adalah berkaitan dengan masa yang digunakan dalam projek ini. Masalah ini timbul kerana saya tidak dapat mengawal masa yang digunakan dalam projek ini. Masalah ini timbul kerana saya tidak dapat mengawal masa yang digunakan dalam projek ini. Masalah ini timbul kerana saya tidak dapat mengawal masa yang digunakan dalam projek ini.

8.0 MASALAH DAN PENYELESAIANNYA

Sepanjang projek ini dibangunkan, pelbagai masalah telah dihadapi tidak kira dalaman atau luaran. Ketika fasa pembangunan permainan beranimasi ini dijalankan, segala masalah yang timbul akan cuba diselesaikan dengan cara yang terbaik dan pantas agar projek ini dapat diteruskan. Berikut merupakan beberapa masalah yang dihadapi ketika pembangunan dijalankan :

1) Kekurangan sumber dan maklumat

- tajuk asal bagi projek ini ialah Permainan Beranimasi Untuk Telefon Bimbit. Oleh kerana kekurangan sumber dan maklumat ke atas kajian permainan telefon bimbit maka projek ini ditukar kepada Permainan Beranimasi Untuk Pocket PC kerana maklumat untuk tajuk ini lebih banyak dan mudah didapati.

2) Kurang penguasaan dalam perisian yang digunakan

- oleh kerana perisian Microsoft Embedded Visual Basic dan C++ tidak diajar di kelas, saya terpaksa belajar sendiri melalui buku dan laman web yang berkenaan. Malahan, subjek mengenai PDA/Pocket PC juga kurang diketengahkan oleh fakulti.

3) Bebanan akademik dan kesuntukan masa

- untuk membina permainan beranimasi memerlukan masa yang agak panjang memandangkan ia bergantung terus kepada pengaturcaraan dan tidak berasaskan web. Selain itu, saya juga mengambil beberapa subjek yang mempunyai tugas yang agak berat. Namun begitu, semua ini dapat diatasi dengan membuat jadual yang ketat dan berdisiplin agar tidak membuang masa terlalu

banyak terhadap perkara yang tak berfaedah dan menumpukan perhatian kepada pembangunan sistem.

4) Mengalami masalah ralat rangkaian

- ralat rangkaian merupakan ralat di mana semua pengkodan dapat dikompil tetapi tidak dapat dilarikan kerana ada beberapa fail rangkaian yang tidak dapat berfungsi dengan baik. Masalah ini diatasi dengan memuat turun beberapa fail rangkaian yang berkaitan dari laman web ke direktori yang berkenaan. Malahan, Cik Siti Hafizah juga memberikan cadangan agar memasuki forum dan 'mIRC' pengaturcara untuk mencari jalan penyelesaian mengenai masalah ini.

8.1 KEKUATAN DAN PERKEMBANGAN PROJEK

Permainan beranimasi yang dibina dalam projek ini dirangkaikan pada fail *GameLibrary* yang merupakan rujukan utama pada sebarang permainan. Kekuatan fail *GameLibrary* ini ialah ia boleh dirangkaikan dengan sebarang permainan beranimasi tidak mengira saiz dan jenis asalkan ia di dalam pelantar Pocket PC. Selain itu, permainan yang dibangunkan juga boleh diubah dan dikembangkan lagi menjadi lebih baik. Contohnya, latar belakang yang lebih baik, komponen permainan lebih menarik, tahap kesusahan ditambah dan lain-lain lagi.

Dengan kata lain, permainan beranimasi ini boleh dihasilkan dengan pelbagai versi dan cara pembinaan. Contohnya, jika projek ini versi 1.0, mungkin permainan ini akan dipertingkatkan pembinaannya dan menjadi versi 1.1 atau 2.0. Ini bergantung kepada citarasa pembangun dan pengguna. Bagi perkembangan permainan beranimasi

ini, segala olahan dan pengubahan boleh dilakukan pada pengkodan projek itu sendiri atau pada *GameLibrary*.

8.2 KESIMPULAN PROJEK

Latihan Ilmiah Tahap 2 merupakan laporan keseluruhan pembangunan beranimasi untuk Pocket PC ini. Ia merangkumi penerangan sepenuhnya beberapa fasa pembangunan sistem seperti objektif, analisis literasi dan metodologi, rekabentuk sistem, implementasi, pengujian, penyeleggaran dan penilaian. Banyak maklumat dan sumber mengenai permainan beranimasi dan Pocket PC dapat diperolehi dari buku rujukan, majalah dan laman web untuk memudahkan kajian.

Tajuk ini dipilih kerana permainan komputer masih belum mendapat sambutan di Malaysia lebih-lebih lagi untuk Pocket PC. Pengaturcaraan dan pengkodannya agak susah berbanding dengan pembangunan laman web. Dari apa yang dijangkakan, permainan beranimasi ini dapat mencapai objektif yang telah diterangkan. Permainan ini dapat dimainkan oleh semua peringkat kerana ia sangat mudah difahami dan mudah untuk dikawal. Malah, pengkodan permainan ini juga dapat diperbaiki lagi ke versi yang lebih kompleks dan lebih menarik.

Tidak ramai pelajar FSKTM yang ingin mengambil tajuk permainan beranimasi sebagai tajuk tesis mereka tetapi dengan penghasilan projek ini diharap agar menjadi rujukan dan panduan untuk lebih ramai lagi pelajar menceburi bidang pembinaan permainan beranimasi. Semoga Permainan Beranimasi Untuk Pocket PC ini dapat digunakan oleh semua golongan terutamanya yang mempunyai Pocket PC dan lebih ramai lagi yang membangunkan permainan beranimasi.

9.9 Kod Summer Tank Battle (Project01)

Project01 - Kod Summer Tank Battle

Project01 - Kod Summer Tank Battle

Project01 - Kod Summer Tank Battle

Project01 - Kod Summer Tank Battle

Project01 - Kod Summer Tank Battle

Project01 - Kod Summer Tank Battle

Project01 - Kod Summer Tank Battle

Project01 - Kod Summer Tank Battle

Project01 - Kod Summer Tank Battle

Project01 - Kod Summer Tank Battle

Project01 - Kod Summer Tank Battle

Project01 - Kod Summer Tank Battle

KOD SUMMER

Project01 - Kod Summer Tank Battle

Project01 - Kod Summer Tank Battle

Project01 - Kod Summer Tank Battle

Project01 - Kod Summer Tank Battle

Project01 - Kod Summer Tank Battle

Project01 - Kod Summer Tank Battle

Project01 - Kod Summer Tank Battle

Project01 - Kod Summer Tank Battle

Project01 - Kod Summer Tank Battle

Project01 - Kod Summer Tank Battle

Project01 - Kod Summer Tank Battle

Project01 - Kod Summer Tank Battle

Project01 - Kod Summer Tank Battle

Project01 - Kod Summer Tank Battle

Project01 - Kod Summer Tank Battle

Project01 - Kod Summer Tank Battle

Project01 - Kod Summer Tank Battle

Project01 - Kod Summer Tank Battle

Project01 - Kod Summer Tank Battle

Project01 - Kod Summer Tank Battle

Project01 - Kod Summer Tank Battle

Project01 - Kod Summer Tank Battle

Project01 - Kod Summer Tank Battle

Project01 - Kod Summer Tank Battle

9.0 Kod Sumber Tank Battle (Project01)

```

////////////////////////////////////
// Pengaturcaraan Permainan Pocket PC
// Kod Sumber Project01.cpp
// Tank Battle
////////////////////////////////////

#include "stdafx.h"
#include "Project01.h"

////////////////////////////////////
// GameInit
////////////////////////////////////
BOOL GameInit(HINSTANCE hInst)
{
    //cipta objek gamelibrary baru
    Game = new CGameLibrary(hInst, _T("Project01"));
    if (Game == NULL)
        return FALSE;

    //sediakan windows untuk penciptaan
    Game->SetFullscreen(TRUE);
    Game->G_SetDisplay(FALSE);
    Game->SetTitle(_T("Tank Battle"));
    Game->SetFrameRate(FRAME_RATE);

    return TRUE;
}

////////////////////////////////////
// GameStart
////////////////////////////////////
void GameStart(HWND hWnd)
{
    srand(GetTickCount());

    //cipta double buffer
    cbDoubleBuffer = new CBitmap(GetDC(hWnd));
    if (!cbDoubleBuffer->Create(Game->ScreenWidth(), Game->ScreenHeight()))
        Game->FatalError(L"Error creating double buffer");

    g_hdc = cbDoubleBuffer->GetSourceDC();

    cbBackground = new CBitmap(g_hdc);
    if (!cbBackground->Load(Game->GetPath(L"background.bmp")))
        Game->FatalError(L"Error loading background.bmp");

    csTiles = new CAnimSprite(cbBackground->GetSourceDC());
    if (!csTiles->LoadAnimSeq(0, 12, 0, Game->GetPath(L"terrain.bmp"), 16, 16))
        Game->FatalError(L"Error loading terrain.bmp");

    for (int y = 0; y < MAP_COLS; y++)
    {
        for (int x = 0; x < MAP_ROWS; x++)
        {
            csTiles->SetFrame(map[y][x]);
            csTiles->SetX(x * 16);
            csTiles->SetY(16 + y * 16);
        }
    }
}

```



```

        csTiles->BitBlit();
    }
}

for (int n = 0; n < 2; n++)
{
    csTank[n] = new CAnimSprite(g_hdc);
    if (!csTank[n]->LoadAnimSeq(0, 8, 0, Game->GetPath(L"tank.bmp"), 16, 16))
        Game->FatalError(L"Error loading tank.bmp");
}

cbTitle = new CBitmap(g_hdc);
if (!cbTitle->Load(Game->GetPath(L"title.bmp")))
    Game->FatalError(L"Error loading title.bmp");

for (n = 0; n < 2; n++)
{
    csShot[n] = new CSprite(g_hdc);
    if (!csShot[n]->Load(Game->GetPath(L"shell.bmp")))
        Game->FatalError(L"Error loading shell.bmp");
}

for (n = 0; n < 3; n++)
{
    csExplosion[n] = new CAnimSprite(g_hdc);
    csExplosion[n]->LoadAnimSeq(0, 8, 0, Game->GetPath(L"explosion.bmp"), 16, 16);
}

cbTitle->BitBlit(0, 0);
}

////////////////////////////////////
// GameActivate
////////////////////////////////////
void GameActivate(HWND hWnd)
{
}

////////////////////////////////////
// ResetGame
////////////////////////////////////
void ResetGame()
{
    int n;

    cbBackground->BitBlit(0, 0);

    pTankPos[0].x = Random() % 150 + 40;
    pTankPos[0].y = 285;
    csTank[0]->SetX(pTankPos[0].x);
    csTank[0]->SetY(pTankPos[0].y);
    csTank[0]->SetXSpeed(0);
    csTank[0]->SetYSpeed(0);
    csTank[0]->SetFrame(0);
    csTank[0]->SetAlive(TRUE);

    pTankPos[1].x = Random() % 150 + 40;
    pTankPos[1].y = 20;
    csTank[1]->SetX(pTankPos[1].x);

```

```
csTank[1]->SetY(pTankPos[1].y);
csTank[1]->SetXSpeed(0);
csTank[1]->SetYSpeed(0);
csTank[1]->SetFrame(4);
csTank[1]->SetAlive(TRUE);
```

```
for (n = 0; n < 2; n++)
```

```
{
    csShot[n]->SetAlive(FALSE);
    csShot[n]->SetX(0);
    csShot[n]->SetY(0);
}
```

```
for (n = 0; n < 3; n++)
```

```
{
    csExplosion[n]->SetFrame(0);
}
```

```
iExplosionIndex = 0;
```

```
bGameOver = FALSE;
```

```
iGameMode = GAME_PLAY;
```

```
}
```

```
////////////////////////////////////
```

```
// GamePaint
```

```
////////////////////////////////////
```

```
void GamePaint(HWND hWnd)
```

```
{
```

```
    HDC hdc;
```

```
    PAINTSTRUCT ps;
```

```
hdc = BeginPaint(hWnd, &ps);
```

```
BitBlt(hdc, 0, 0, Game->ScreenWidth(), Game->ScreenHeight(),
```

```
    cbDoubleBuffer->GetSourceDC(), 0, 0, SRCCOPY);
```

```
EndPaint(hWnd, &ps);
```

```
}
```

```
////////////////////////////////////
```

```
// GameEvent
```

```
////////////////////////////////////
```

```
void GameEvent()
```

```
{
```

```
    HWND hWnd;
```

```
    HDC hdc;
```

```
hWnd = Game->GetWindow();
```

```
hdc = GetDC(hWnd);
```

```
if (!bGameOver && iGameMode == GAME_PLAY)
```

```
{
```

```
    Animate(g_hdc);
```

```
}
```

```
BitBlt(hdc, 0, 0, Game->ScreenWidth(), Game->ScreenHeight(),
```

```
    g_hdc, 0, 0, SRCCOPY);
```

```
ReleaseDC(hWnd, hdc);
```

```
}
```



```

////////////////////////////////////
// StylusDown
////////////////////////////////////
void StylusDown(int x, int y)
{
    POINT pt;
    pt.x = x;
    pt.y = y;
    switch (iGameMode)
    {
        case GAME_TITLE:
            if (Game->PointInside(pt, rectStart))
            {
                ResetGame();
            }
            else if (Game->PointInside(pt, rectQuit))
            {
                Game->Shutdown();
            }
            break;

        case GAME_PLAY:
            iGameMode = GAME_TITLE;
            bGameOver = TRUE;
            cbTitle->BitBlit(0, 0);
            break;
    }
}

```

```

////////////////////////////////////
// StylusMove
////////////////////////////////////
void StylusMove(int x, int y)
{
}

```

```

////////////////////////////////////
// StylusUp
////////////////////////////////////
void StylusUp(int x, int y)
{
}

```

```

////////////////////////////////////
// ButtonPress
////////////////////////////////////
void ButtonPress(int iButtonID, POINT pt)
{
    static unsigned int iTimer = 0;
    int speed = csTank[0]->GetXSpeed();

    if (iGameMode == GAME_PLAY)
    {
        switch (iButtonID)
        {
            case 0: //UP
                if (speed < 1)
                {
                    csTank[0]->SetXSpeed(speed + 1);
                    break;
                }
            }
        }
    }
}

```

```

case 1: //DOWN
    if (speed > -1)
        csTank[0]->SetXSpeed(speed - 1);
    break;

```

```

case 2: //LEFT
    if (iTimer + 500 < GetTickCount())
    {
        csTank[0]->PrevFrame();
        iTimer = GetTickCount();
    }
    break;

```

```

case 3: //RIGHT
    if (iTimer + 500 < GetTickCount())
    {
        csTank[0]->NextFrame();
        iTimer = GetTickCount();
    }
    break;

```

```

case 4:

```

```

case 5:

```

```

case 6:

```

```

    FireShot(0);
    break;

```

```

}
}
}

```

```

// ButtonRelease

```

```

// ButtonRelease

```

```

void ButtonRelease(int iButtonID, POINT pt)

```

```

{

```

```

    switch (iButtonID)
    {

```

```

        case 2: //LEFT

```

```

        case 3: //RIGHT

```

```

            csTank[0]->SetXSpeed(0);

```

```

            break;

```

```

        }
    }
}

```

```

// GameEnd

```

```

// GameEnd

```

```

// GameEnd

```

```

// GameEnd

```

```

// GameEnd

```

```

// GameEnd

```

```

// GameEnd

```

```

// GameEnd

```

```

// GameEnd

```

```

// GameEnd

```

```

// GameEnd

```

```

// GameEnd

```

```

// GameEnd

```

```

// GameEnd

```

```

// GameEnd

```

```

// GameEnd

```

```

// GameEnd

```

```

// GameEnd

```

```

// GameEnd

```

```

// GameEnd

```

```

// GameEnd

```

```

// GameEnd

```

```

// GameEnd

```



```

delete csExplosion[1];
delete csExplosion[2];
}

//////////
// Animate
//////////
void Animate(HDC hdc)
{
    static BOOL bFirst = TRUE;
    BOOL bAlive = FALSE;
    int n;

    if (bFirst)
    {
        csTank[0]->SaveUnder(hdc);
        csTank[1]->SaveUnder(hdc);
        for (n = 0; n < 2; n++)
            csShot[n]->SaveUnder(hdc);
        for (n = 0; n < 3; n++)
            csExplosion[n]->SaveUnder(hdc);
        bFirst = FALSE;
    }

    BitBlt(hdc, 0, 0, 239, 16, cbBackground->GetSourceDC(), 0, 0, SRCCOPY);
    BitBlt(hdc, 0, 301, 239, 16, cbBackground->GetSourceDC(), 0, 301, SRCCOPY);

    for (n = 0; n < 2; n++)
        csShot[n]->RestoreUnder(hdc);

    for (n = 0; n < 3; n++)
        csExplosion[n]->RestoreUnder(hdc);

    csTank[0]->RestoreUnder(hdc);
    csTank[1]->RestoreUnder(hdc);

    for (n = 0; n < 2; n++)
    {
        if (csShot[n]->IsAlive())
        {
            UpdateShotPosition(n);
            csShot[n]->SaveUnder(hdc);
        }
    }

    ManeuverTank(1, 0);

    for (n = 0; n < 2; n++)
    {
        UpdateTankPosition(n);
        csTank[n]->SaveUnder(hdc);
    }

    for (n = 0; n < 2; n++)
    {
        if (csShot[n]->IsAlive())
        {
            csShot[n]->TransBlit(BLACK);

```

```

    }
}

csTank[0]->TransBlit(BLACK);
csTank[1]->TransBlit(BLACK);

for (n = 0; n < 3; n++)
{
    if (csExplosion[n]->IsAlive())
    {
        csExplosion[n]->TransBlit(BLACK);
        if (csExplosion[n]->GetFrame() < 7)
            csExplosion[n]->NextFrame();
        else
        {
            csExplosion[n]->SetAlive(FALSE);
        }
    }
}

wsprintf(tcScore, L"TANK: %i/%i/%i", csTank[0]->GetFrame(), pTankPos[0].x, pTankPos[0].y);
Game->PrintTextXY(hdc, tcScore, 1, 303, RED, TRANSPARENT);

}

////////////////////////////////////
// UpdateTankPosition
////////////////////////////////////
void UpdateTankPosition(int iTank)
{
    switch (csTank[iTank]->GetFrame())
    {
        case 0:
            pTankPos[iTank].y -= csTank[iTank]->GetXSpeed();
            break;
        case 1:
            pTankPos[iTank].y -= csTank[iTank]->GetXSpeed();
            pTankPos[iTank].x += csTank[iTank]->GetXSpeed();
            break;
        case 2:
            pTankPos[iTank].x += csTank[iTank]->GetXSpeed();
            break;
        case 3:
            pTankPos[iTank].y += csTank[iTank]->GetXSpeed();
            pTankPos[iTank].x += csTank[iTank]->GetXSpeed();
            break;
        case 4:
            pTankPos[iTank].y += csTank[iTank]->GetXSpeed();
            break;
        case 5:
            pTankPos[iTank].y += csTank[iTank]->GetXSpeed();
            pTankPos[iTank].x -= csTank[iTank]->GetXSpeed();
            break;
        case 6:
            pTankPos[iTank].x -= csTank[iTank]->GetXSpeed();
            break;
        case 7:
            pTankPos[iTank].y -= csTank[iTank]->GetXSpeed();
            pTankPos[iTank].x -= csTank[iTank]->GetXSpeed();
            break;
    }
}

```



```

    }
    if (pTankPos[iTank].x < 5)
        pTankPos[iTank].x = 5;
    else if (pTankPos[iTank].x > 220)
        pTankPos[iTank].x = 220;
    if (pTankPos[iTank].y < 18)
        pTankPos[iTank].y = 18;
    else if (pTankPos[iTank].y > 285)
        pTankPos[iTank].y = 285;
    csTank[iTank]->SetX(pTankPos[iTank].x);
    csTank[iTank]->SetY(pTankPos[iTank].y);
}

```

```

////////////////////////////////////

```

```

// PolarTrajectory

```

```

////////////////////////////////////

```

```

int PolarTrajectory(int iTraj)

```

```

{

```

```

    switch (iTraj)
    {

```

```

        case 0:

```

```

            return 4;

```

```

        case 1:

```

```

            return 5;

```

```

        case 2:

```

```

            return 6;

```

```

        case 3:

```

```

            return 7;

```

```

        case 4:

```

```

            return 0;

```

```

        case 5:

```

```

            return 1;

```

```

        case 6:

```

```

            return 2;

```

```

        case 7:

```

```

            return 3;

```

```

    }

```

```

}

```

```

////////////////////////////////////

```

```

// ManeuverTank

```

```

////////////////////////////////////

```

```

void ManeuverTank(int iTank, int iEnemy)

```

```

{

```

```

    static POINT pLastShot = {0, 0};

```

```

    static int iCompDir[2];

```

```

    static int iCompDist[2];

```

```

    POINT shot;

```

```

    TCHAR szTemp[40];

```

```

    int iTraj;

```

```

    shot.x = csShot[iEnemy]->GetX();

```

```

    shot.y = csShot[iEnemy]->GetY();

```

```

    iTraj = CalculateTrajectory(shot, pLastShot);

```

```

    wsprintf(szTemp, L"Traj: %i", iTraj);

```

```

    Game->PrintTextXY(g_hdc, szTemp, 1, 1, WHITE, TRANSPARENT);

```

```

    pLastShot.x = shot.x;

```

```

    pLastShot.y = shot.y;

```

```

if (CalculateDanger(iTank, shot, iTraj))
{
    wsprintf(szTemp, L"Danger: %i/%i", shot.x, shot.y);
    Game->PrintTextXY(g_hdc, szTemp, 145, 1, RED, TRANSPARENT);

    switch (iTraj)
    {
        case 0:
        case 4:
            while (iCompDir[iTank] == 4 || iCompDir[iTank] == 0)
                iCompDir[iTank] = Random() % 8;
            break;
        case 1:
        case 5:
            while (iCompDir[iTank] == 5 || iCompDir[iTank] == 1)
                iCompDir[iTank] = Random() % 8;
            break;
        case 2:
        case 6:
            while (iCompDir[iTank] == 6 || iCompDir[iTank] == 2)
                iCompDir[iTank] = Random() % 8;
            break;
        case 3:
        case 7:
            while (iCompDir[iTank] == 7 || iCompDir[iTank] == 3)
                iCompDir[iTank] = Random() % 8;
            break;
    }

    iCompDist[iTank] = 10;
    csTank[iTank]->SetXSpeed(1);
    csTank[iTank]->SetFrame(iCompDir[iTank]);
}
else
{
    iTraj = CalculateTrajectory(pTankPos[iTank], pTankPos[iEnemy]);
    if (CalculateDanger(iTank, pTankPos[iEnemy], iTraj))
    {
        wsprintf(szTemp, L"Target: %i/%i", pTankPos[iEnemy].x, pTankPos[iEnemy].y);
        Game->PrintTextXY(g_hdc, szTemp, 145, 1, GREEN, TRANSPARENT);

        csTank[iTank]->SetXSpeed(0);
        csTank[iTank]->SetFrame(PolarTrajectory(iTraj));
        FireShot(iTank);
    }
    else
    {
        if (pTankPos[iTank].x < 30 || pTankPos[iTank].x > 210 ||
            pTankPos[iTank].y < 30 || pTankPos[iTank].y > 280)
            iCompDir[iTank] = PolarTrajectory(iTraj);
        else if (iCompDist[iTank]-- < 0)
        {
            iCompDir[iTank] = Random() % 8;
            iCompDist[iTank] = Random() % 10 + 5;
        }
        csTank[iTank]->SetFrame(iCompDir[iTank]);
        csTank[iTank]->SetXSpeed(1);
    }
}
}

```



```

}

// CalculateTrajectory
// CalculateTrajectory(POINT A, POINT B)
{
    int iTraj = -1;

    if (A.x == B.x && A.y < B.y)
        iTraj = 0;
    else if (A.x == B.x && A.y > B.y)
        iTraj = 4;

    if (A.y == B.y && A.x < B.x)
        iTraj = 6;
    else if (A.y == B.y && A.x > B.x)
        iTraj = 2;

    if (A.x > B.x && A.y < B.y)
        iTraj = 1;
    else if (A.x < B.x && A.y < B.y)
        iTraj = 7;

    if (A.y > B.y && A.x > B.x)
        iTraj = 3;
    else if (A.y > B.y && A.x < B.x)
        iTraj = 5;

    return iTraj;
}

// CalculateDanger
// CalculateDanger(int iTank, POINT pShot, int iTraj)
{
    RECT tank;
    static int iCompDir = 0;
    static int iCompDist = 0;

    tank.left = csTank[iTank]->GetX();
    tank.top = csTank[iTank]->GetY();
    tank.right = tank.left + csTank[iTank]->ImageWidth();
    tank.bottom = tank.top + csTank[iTank]->ImageHeight();

    switch (iTraj)
    {
        case 0:
            if (pTankPos[iTank].y < pShot.y &&
                pTankPos[iTank].x - 4 < pShot.x &&
                pTankPos[iTank].x + 14 > pShot.x)
                return TRUE;
            break;

        case 1:
            if (pTankPos[iTank].x > pShot.x && pTankPos[iTank].y < pShot.y)
            {
                while (pShot.x < 240 && pShot.y > 0)
            {

```

```

        if (Game->PointInside(pShot, tank))
            return TRUE;
        pShot.x++;
        pShot.y--;
    }
}
return FALSE;
break;

case 2:
    if (pTankPos[iTank].x > pShot.x &&
        pTankPos[iTank].y - 4 < pShot.y &&
        pTankPos[iTank].y + 14 > pShot.y)
        return TRUE;
    break;

case 3:
    if (pTankPos[iTank].x > pShot.x && pTankPos[iTank].y > pShot.y)
    {
        while (pShot.x < 240 && pShot.y < 320)
        {
            if (Game->PointInside(pShot, tank))
                return TRUE;
            pShot.x++;
            pShot.y++;
        }
    }
    return FALSE;
    break;

case 4:
    if (pTankPos[iTank].y > pShot.y &&
        pTankPos[iTank].x - 4 < pShot.x &&
        pTankPos[iTank].x + 14 > pShot.x)
        return TRUE;
    break;

case 5:
    if (pTankPos[iTank].x < pShot.x && pTankPos[iTank].y > pShot.y)
    {
        while (pShot.x > 0 && pShot.y < 320)
        {
            if (Game->PointInside(pShot, tank))
                return TRUE;
            pShot.x--;
            pShot.y++;
        }
    }
    return FALSE;
    break;

case 6:
    if (pTankPos[iTank].x < pShot.x &&
        pTankPos[iTank].y - 4 < pShot.y &&
        pTankPos[iTank].y + 14 > pShot.y)
        return TRUE;
    break;

case 7:
    if (pTankPos[iTank].x < pShot.x && pTankPos[iTank].y < pShot.y)
    {
        while (pShot.x > 0 && pShot.y > 0)
        {
            if (Game->PointInside(pShot, tank))
                return TRUE;

```



```

        pShot.x--;
        pShot.y--;
    }
}
return FALSE;
break;
}

return FALSE;
}

////////////////////////////////////
// FireShot
////////////////////////////////////
void FireShot(int iTank)
{
    if (csShot[iTank]->IsAlive())
        return;

    csShot[iTank]->SetAlive(TRUE);

    switch (csTank[iTank]->GetFrame())
    {
        case 0: //NORTH
            csShot[iTank]->SetX(pTankPos[iTank].x + 5);
            csShot[iTank]->SetY(pTankPos[iTank].y - 6);
            csShot[iTank]->SetXSpeed(0);
            csShot[iTank]->SetYSpeed(-SHOT_SPEED);
            break;
        case 1: //NE
            csShot[iTank]->SetX(pTankPos[iTank].x + 13);
            csShot[iTank]->SetY(pTankPos[iTank].y - 2);
            csShot[iTank]->SetXSpeed(SHOT_SPEED);
            csShot[iTank]->SetYSpeed(-SHOT_SPEED);
            break;
        case 2: //EAST
            csShot[iTank]->SetX(pTankPos[iTank].x + 13);
            csShot[iTank]->SetY(pTankPos[iTank].y + 3);
            csShot[iTank]->SetXSpeed(SHOT_SPEED);
            csShot[iTank]->SetYSpeed(0);
            break;
        case 3: //SE
            csShot[iTank]->SetX(pTankPos[iTank].x + 13);
            csShot[iTank]->SetY(pTankPos[iTank].y + 13);
            csShot[iTank]->SetXSpeed(SHOT_SPEED);
            csShot[iTank]->SetYSpeed(SHOT_SPEED);
            break;
        case 4: //SOUTH
            csShot[iTank]->SetX(pTankPos[iTank].x + 5);
            csShot[iTank]->SetY(pTankPos[iTank].y + 13);
            csShot[iTank]->SetXSpeed(0);
            csShot[iTank]->SetYSpeed(SHOT_SPEED);
            break;
        case 5: //SW
            csShot[iTank]->SetX(pTankPos[iTank].x - 4);
            csShot[iTank]->SetY(pTankPos[iTank].y + 10);
            csShot[iTank]->SetXSpeed(-SHOT_SPEED);
            csShot[iTank]->SetYSpeed(SHOT_SPEED);
            break;
        case 6: //WEST

```

```

        csShot[iTank]->SetX(pTankPos[iTank].x - 4);
        csShot[iTank]->SetY(pTankPos[iTank].y + 3);
        csShot[iTank]->SetXSpeed(-SHOT_SPEED);
        csShot[iTank]->SetYSpeed(0);
        break;
    case 7: //NW
        csShot[iTank]->SetX(pTankPos[iTank].x - 4);
        csShot[iTank]->SetY(pTankPos[iTank].y - 4);
        csShot[iTank]->SetXSpeed(-SHOT_SPEED);
        csShot[iTank]->SetYSpeed(-SHOT_SPEED);
        break;
}

csTank[iTank]->RestoreUnder(g_hdc);
csShot[iTank]->SaveUnder(g_hdc);
}

////////////////////////////////////
// UpdateShotPosition
////////////////////////////////////
void UpdateShotPosition(int iShot)
{
    POINT pShotPos;
    POINT pt;
    int n;

    pShotPos.x = csShot[iShot]->GetX() + csShot[iShot]->GetXSpeed();
    pShotPos.y = csShot[iShot]->GetY() + csShot[iShot]->GetYSpeed();

    if (pShotPos.x < 4)
        csShot[iShot]->SetAlive(FALSE);
    else if (pShotPos.x > 230)
        csShot[iShot]->SetAlive(FALSE);
    if (pShotPos.y < 18)
        csShot[iShot]->SetAlive(FALSE);
    else if (pShotPos.y > 297)
        csShot[iShot]->SetAlive(FALSE);

    csShot[iShot]->SetX(pShotPos.x);
    csShot[iShot]->SetY(pShotPos.y);

    for (n = 0; n < 2; n++)
    {
        if (csShot[iShot]->Intersected(csTank[n], pt))
        {
            csShot[iShot]->SetAlive(FALSE);
            csShot[iShot]->SetX(0);
            csShot[iShot]->SetY(0);
            csTank[n]->SetAlive(FALSE);
            csExplosion[n]->SetX(csTank[n]->GetX());
            csExplosion[n]->SetY(csTank[n]->GetY());
            csExplosion[n]->SetAlive(TRUE);
            csExplosion[n]->SetFrame(0);
            csExplosion[n]->SaveUnder(g_hdc);
        }
    }
}

```



```

////////////////////////////////////
// Pengaturcaraan Permainan Pocket PC
// Kod Sumber Project01.h
// Tank Battle
////////////////////////////////////

```

```
#pragma once
```

```

#define WHITE RGB(255,255,255)
#define BLACK RGB(0,0,0)
#define RED RGB(255,0,0)
#define GREEN RGB(0,255,0)
#define BLUE RGB(0,0,255)

```

```

#define GAME_TITLE 0
#define GAME_PLAY 2

```

```

void Animate(HDC hdc);
void ResetGame();
void FireShot(int iTank);
void UpdateShotPosition(int iShot);
void UpdateTankPosition(int iTank);
void ManeuverTank(int iTank, int iEnemy);
int CalculateTrajectory(POINT A, POINT B);
BOOL CalculateDanger(int iTank, POINT pShot, int iTraj);

```

```
#define FRAME_RATE 20
```

```
#define SHOT_SPEED 4
```

```

RECT rectStart = {40, 100, 200, 140};
RECT rectQuit = {40, 190, 200, 230};

```

```

#define MAP_COLS 18
#define MAP_ROWS 15
int map[MAP_COLS][MAP_ROWS] =
{{8,1,1,1,1,1,1,1,1,1,1,1,1,9 },
{6,0,0,0,0,0,0,0,0,0,0,0,0,0,3 },
{6,0,0,0,0,0,0,0,0,0,0,0,0,0,3 },
{6,0,2,2,2,0,0,0,0,0,0,0,2,2,3 },
{6,0,2,2,2,0,0,0,0,0,0,0,2,2,3 },
{6,0,2,2,0,0,0,0,0,0,0,0,0,0,3 },
{6,0,0,0,0,0,0,0,0,0,0,0,0,0,3 },
{6,0,0,0,0,0,0,0,0,0,0,0,0,0,3 },
{6,0,0,0,0,0,0,0,0,0,2,2,0,0,3 },
{6,0,0,0,0,0,0,0,0,0,2,2,0,0,3 },
{6,0,0,2,2,0,0,0,0,0,2,2,0,0,3 },
{6,0,0,2,2,0,0,0,0,0,0,0,0,0,3 },
{6,0,0,2,2,0,0,0,0,0,0,0,0,0,3 },
{6,0,0,0,0,0,0,0,0,0,0,0,2,2,0,3 },
{6,0,0,0,0,0,0,0,0,0,0,0,2,2,3 },
{6,0,0,0,0,0,0,0,0,0,0,0,0,0,3 },
{10,5,5,5,5,5,5,5,5,5,5,5,5,11 } };

```

```
CGameLibrary *Game;
```

```
CBitmap *cbDoubleBuffer;
```

```
CBitmap *cbBackground;  
CBitmap *cbTitle;  
CAnimSprite *csTiles;  
CAnimSprite *csTank[2];  
CSprite *csShot[2];  
CAnimSprite *csEnemyTankLeft;  
CAnimSprite *csExplosion[3];
```

```
HDC g_hdc;  
POINT pTankPos[2];  
TCHAR tcScore[20];  
int iExplosionIndex = 0;  
int iGameMode = GAME_TITLE;  
BOOL bGameOver = FALSE;
```

University of Malaya

9.1 Kod Sumber GameLibrary

```

////////////////////////////////////
// Pengaturcaraan Permainan Pocket PC
// Kod Sumber CGameLibrary
////////////////////////////////////

#include "stdafx.h"
#include "GameLibrary.h"

CGameLibrary *CGameLibrary::pGameLib = NULL;

////////////////////////////////////
// CGameLibrary::CGameLibrary constructor
////////////////////////////////////
CGameLibrary::CGameLibrary(HINSTANCE hInst, LPTSTR szNewWindowClass)
{
    pGameLib = this;
    bHibernate = TRUE;
    bFullscreen = FALSE;
    hInstance = hInst;
    hWindow = NULL;
    iFrameRate = 60;
    bGAPIDisplay = FALSE;

    if (wcslen(szNewWindowClass) > 0)
        wcsncpy(szWindowClass, szNewWindowClass);
    else
        wcsncpy(szWindowClass, _T("<Unnamed Window Class>"));

    if (wcslen(szTitle) == 0)
        wcsncpy(szTitle, _T("<Unnamed Title>"));
}

////////////////////////////////////
// CGameLibrary::~CGameLibrary destructor
////////////////////////////////////
CGameLibrary::~CGameLibrary()
{
}

////////////////////////////////////
// CGameLibrary::Initialize
////////////////////////////////////
BOOL CGameLibrary::Initialize(int nCmdShow)
{
    WNDCLASS wc;
    RECT rect;

    hWindow = FindWindow(szWindowClass, szTitle);
    if (hWindow)
    {
        SetForegroundWindow ((HWND) (((DWORD)hWindow) | 0x01));
        return 0;
    }

    wc.style          = CS_HREDRAW | CS_VREDRAW;
    wc.lpfnWndProc    = (WNDPROC) WndProc;
    wc.cbClsExtra     = 0;

```

```

wc.cbWndExtra          = 0;
wc.hInstance           = GetInstance();
wc.hIcon               = LoadIcon(hInstance, MAKEINTRESOURCE(GetIcon()));
wc.hCursor             = 0;
wc.hbrBackground       = (HBRUSH) GetStockObject(WHITE_BRUSH);
wc.lpszMenuName        = 0;
wc.lpszClassName       = szWindowClass;

if (!RegisterClass(&wc))
{
    Error(_T("Could not register window class!"));
}

GetClientRect(hWindow, &rect);

if (GetFullscreen() || G_Enabled())
{
    hWindow = CreateWindow(szWindowClass, szTitle,
        WS_VISIBLE, 0, 0,
        GetSystemMetrics(SM_CXSCREEN),
        GetSystemMetrics(SM_CYSCREEN),
        NULL, NULL, hInstance, NULL);
} else {
    hWindow = CreateWindow(szWindowClass, szTitle,
        WS_VISIBLE, CW_USEDEFAULT, CW_USEDEFAULT, CW_USEDEFAULT,
        CW_USEDEFAULT, NULL, NULL, hInstance, NULL);
}

if (!hWindow)
{
    Error(_T("Could not create the game window!"));
    return FALSE;
}

ShowWindow(hWindow, nCmdShow);
UpdateWindow(hWindow);

if (G_Enabled())
{
    if (GXOpenDisplay(GetWindow(), GX_FULLSCREEN) == 0) {
        return FALSE;
    }

    gxDisplay = GXGetDisplayProperties();

    if (((GetBitsPerPixel() != 16) || (!IsScreenFormat565())))
    {
        Error(_T("Full 16-bit color display is required!"));
        GXCloseDisplay();
        return FALSE;
    }
}

GXOpenInput();
gxKeys = GXGetDefaultKeys(GX_NORMALKEYS);

return TRUE;
}

```



```

////////////////////////////////////
// CGameLibrary::EventHandler
////////////////////////////////////
LRESULT CGameLibrary::EventHandler(HWND hWnd, UINT message, WPARAM wParam,
LPARAM lParam)
{
    short vkKey;
    POINT pt;
    BOOL bButton;

    switch (message)
    {
        case WM_CREATE:
            SetWindow(hWnd);
            SetForegroundWindow(GetWindow());
            GameStart(hWnd);
            break;

            case WM_HIBERNATE:
                SetHibernate(TRUE);
                SHFullScreen(GetWindow(),
SHFS_SHOWSTARTICON|SHFS_SHOWTASKBAR|SHFS_SHOWSIPBUTTON);
                break;

        case WM_KILLFOCUS:
            SetHibernate(TRUE);
            if (G_Enabled())
                GXSuspend();
            break;

        case WM_ACTIVATE:
            GameActivate(hWnd);
            if (GetHibernate())
            {
                SetHibernate(FALSE);
            }
            if (G_Enabled())
                GXResume();
            else
            {
                if (GetFullscreen())
                {
                    if( wParam == WA_INACTIVE )
                        SHFullScreen(hWnd,
SHFS_SHOWSTARTICON|SHFS_SHOWTASKBAR|SHFS_SHOWSIPBUTTON);
                    else
                        SHFullScreen(hWnd,
SHFS_HIDESTARTICON|SHFS_HIDETASKBAR|SHFS_HIDESIPBUTTON);
                }
            }

            break;

        case WM_PAINT:
            GamePaint(hWnd);
            break;

            case WM_LBUTTONDOWN:
                StylusDown(LOWORD(lParam), HIWORD(lParam));
                break;

        case WM_MOUSEMOVE:
    
```

```
StylusMove(LOWORD(lParam), HIWORD(lParam));
break;

    case WM_LBUTTONDOWN:
        StylusUp(LOWORD(lParam), HIWORD(lParam));
        break;

case WM_KEYDOWN:
    vkKey = (short)wParam;
    bButton = TRUE;

    if (vkKey == gxKeys.vkUp)
    {
        pt = gxKeys.ptUp;
        vkKey = 0;
    }

    else if (vkKey == gxKeys.vkDown)
    {
        pt = gxKeys.ptDown;
        vkKey = 1;
    }

    else if (vkKey == gxKeys.vkLeft)
    {
        pt = gxKeys.ptLeft;
        vkKey = 2;
    }

    else if (vkKey == gxKeys.vkRight)
    {
        pt = gxKeys.ptRight;
        vkKey = 3;
    }

    else if (vkKey == gxKeys.vkA)
    {
        pt = gxKeys.ptA;
        vkKey = 4;
    }

    else if (vkKey == gxKeys.vkB)
    {
        pt = gxKeys.ptB;
        vkKey = 5;
    }

    else if (vkKey == gxKeys.vkC)
    {
        pt = gxKeys.ptC;
        vkKey = 6;
    }

    else if (vkKey == gxKeys.vkStart)
    {
        pt = gxKeys.ptStart;
        vkKey = 7;
    }

    else if (vkKey == 192)
```



```
{
    pt = gxKeys.ptStart;
    vkKey = 8;
}

else if (vkKey == 193)
{
    pt = gxKeys.ptStart;
    vkKey = 9;
}

else
    bButton = FALSE;

if (bButton)
    ButtonPress(vkKey, pt);

break;

case WM_KEYUP:
    vkKey = (short)wParam;
    bButton = TRUE;

    if (vkKey == gxKeys.vkUp)
    {
        pt = gxKeys.ptUp;
        vkKey = 0;
    }

    else if (vkKey == gxKeys.vkDown)
    {
        pt = gxKeys.ptDown;
        vkKey = 1;
    }

    else if (vkKey == gxKeys.vkLeft)
    {
        pt = gxKeys.ptLeft;
        vkKey = 2;
    }

    else if (vkKey == gxKeys.vkRight)
    {
        pt = gxKeys.ptRight;
        vkKey = 3;
    }

    else if (vkKey == gxKeys.vkA)
    {
        pt = gxKeys.ptA;
        vkKey = 4;
    }

    else if (vkKey == gxKeys.vkB)
    {
        pt = gxKeys.ptB;
        vkKey = 5;
    }

    else if (vkKey == gxKeys.vkC)
```

```

    {
        pt = gxKeys.ptC;
        vkKey = 6;
    }

    else if (vkKey == gxKeys.vkStart)
    {
        pt = gxKeys.ptStart;
        vkKey = 7;
    }

    else if (vkKey == 192)
    {
        pt = gxKeys.ptStart;
        vkKey = 8;
    }

    else if (vkKey == 193)
    {
        pt = gxKeys.ptStart;
        vkKey = 9;
    }

    else
        bButton = FALSE;

    if (bButton)
        ButtonRelease(vkKey, pt);

    break;

case WM_DESTROY:
    GameEnd();
    if (G_Enabled())
        GXCloseDisplay();

    GXCloseInput();

    PostQuitMessage(0);
    break;

default:
    return DefWindowProc(hWnd, message, wParam, lParam);
}

return 0;
}

////////////////////////////////////
// CGameLibrary::Error
////////////////////////////////////
void CGameLibrary::Error(LPTSTR szError)
{
    TCHAR szMsg[100];

    wsprintf(szMsg, _T("%s:\n%s\n%s, %i"), szMsg, __FILE__, __LINE__);
    MsgBox(szMsg);
}

```



```

////////////////////////////////////
// CGameLibrary::Shutdown
////////////////////////////////////
void CGameLibrary::Shutdown()
{
    PostMessage(hWindow, WM_CLOSE, 0, 0);
}

////////////////////////////////////
// CGameLibrary::G_BeginDraw
////////////////////////////////////
void CGameLibrary::G_BeginDraw()
{
    if (G_Enabled())
        GAPIVidMem = (unsigned char *)GXBeginDraw();
}

////////////////////////////////////
// CGameLibrary::G_EndDraw
////////////////////////////////////
void CGameLibrary::G_EndDraw()
{
    if (G_Enabled())
        GXEndDraw();
}

////////////////////////////////////
// Fungsi : G_ClearScreen
////////////////////////////////////
BOOL CGameLibrary::G_ClearScreen(COLOR color)
{
    G_BeginDraw();
    for (int y = 0; y < ScreenHeight(); y++) {
        for (int x = 0; x < ScreenWidth(); x++) {
            G_DrawPixel16(G_GetVidMem(), x, y, color);
        }
    }
    G_EndDraw();
    return TRUE;
}

////////////////////////////////////
// Fungsi : G_DrawPixel16
////////////////////////////////////
int CGameLibrary::G_DrawPixel16(unsigned char *VidMem, int X, int Y, COLOR color)
{
    unsigned short usColor;
    int address;

    address = (X * GetXPitch()) + (Y * GetYPitch());

    usColor = (unsigned short)
        (((color.Red & 0xf8) << 8) |
         ((color.Green & 0xfc) << 3) |
         ((color.Blue & 0xf8) >> 3));

    *(unsigned short *) (VidMem + address) = usColor;

    return 0;
}

```

```
////////////////////////////////////
```

```
// Fungsi : G_DrawSolidRect
```

```
////////////////////////////////////
```

```
void CGameLibrary::G_DrawSolidRect(unsigned char *VidMem, int iLeft, int iTop, int iRight, int iBottom, COLOR color)
```

```
{
    int x, y;

    for (y = iTop; y<=iBottom; y++)
    {
        for (x = iLeft; x<=iRight; x++)
        {
            G_DrawPixel16(VidMem, x, y, color);
        }
    }
}
```

```
////////////////////////////////////
```

```
// FatalError
```

```
////////////////////////////////////
```

```
void CGameLibrary::FatalError(LPTSTR lpStr)
```

```
{
    MessageBox(GetWindow(), lpStr, _T("Fatal Error"), MB_OK | MB_ICONERROR);
    Shutdown();
}
```

```
////////////////////////////////////
```

```
// PrintText
```

```
////////////////////////////////////
```

```
void CGameLibrary::PrintText(HDC hdc, LPCTSTR strText, int row, int col, COLORREF color, int iBkMode)
```

```
{
    static TEXTMETRIC tm;
    static int cxChar, cyChar, cxCaps;
    static BOOL bFirst = TRUE;

    if (bFirst)
    {
        GetTextMetrics(hdc, &tm);
        cxChar = tm.tmAveCharWidth;
        cyChar = tm.tmHeight + tm.tmExternalLeading;
        cxCaps = (tm.tmPitchAndFamily & 1 ? 3 : 2) * cxChar / 2;
        bFirst = FALSE;
    }
}
```

```
SetTextColor(hdc, color);
SetBkMode(hdc, iBkMode);
ExtTextOut(hdc, row + cxChar, cyChar * col, 0, NULL, strText, wcslen(strText), NULL);
}
```

```
////////////////////////////////////
```

```
// PrintTextXY
```

```
////////////////////////////////////
```

```
void CGameLibrary::PrintTextXY(HDC hdc, LPCTSTR strText, int x, int y, COLORREF color, int iBkMode)
```

```
{
    SetTextColor(hdc, color);
    SetBkMode(hdc, iBkMode);
}
```



```
ExtTextOut(hdc, x, y, 0, NULL, strText, wcslen(strText), NULL);
```

```
}
```

```
//////////////////////////////////////  
// GetPath
```

```
//////////////////////////////////////
```

```
LPWSTR CGameLibrary::GetPath(TCHAR *filename)
```

```
{
```

```
    TCHAR *lpFilename;
```

```
    TCHAR *pDest;
```

```
    int nStrLen;
```

```
    int result;
```

```
    int ch = '\\';
```

```
    lpFilename = new TCHAR[255];
```

```
    nStrLen = GetModuleFileName(NULL, lpFilename, 255);
```

```
    pDest = wcsrchr(lpFilename, ch);
```

```
    if (pDest != NULL)
```

```
    {
```

```
        result = pDest - lpFilename + 1;
```

```
        lpFilename[result] = '\0';
```

```
        if (filename != NULL)
```

```
            wscat(lpFilename, filename);
```

```
        return lpFilename;
```

```
    }
```

```
    else
```

```
        return NULL;
```

```
}
```

```
//////////////////////////////////////  
// Delay
```

```
//////////////////////////////////////
```

```
void CGameLibrary::Delay(int ms)
```

```
{
```

```
    unsigned long iStart = GetTickCount();
```

```
    while (GetTickCount() < iStart + ms);
```

```
}
```

```
//////////////////////////////////////  
// PointInside
```

```
//////////////////////////////////////
```

```
BOOL CGameLibrary::PointInside(POINT pt, int left, int top, int right, int bottom)
```

```
{
```

```
    if (pt.x >= left && pt.x <= right &&
```

```
        pt.y >= top && pt.y <= bottom)
```

```
        return TRUE;
```

```
    else
```

```
        return FALSE;
```

```
}
```

```
//////////////////////////////////////  
// PointInside (overloaded)
```

```
//////////////////////////////////////
```

```
BOOL CGameLibrary::PointInside(POINT pt, RECT rt)
```

```
{
```

```
    return PointInside(pt, rt.left, rt.top, rt.right, rt.bottom);
```

```
}
```

```

////////////////////////////////////
// WndProc
////////////////////////////////////
LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM
lParam)
{
    return GameLib()->EventHandler(hWnd, message, wParam, lParam);
}

////////////////////////////////////
// WinMain
////////////////////////////////////
int WINAPI WinMain(    HINSTANCE hInstance,
                      HINSTANCE hPrevInstance,
                      LPTSTR lpCmdLine,
                      int nCmdShow )
{
    MSG msg;
    static iNewTime = 0;
    int iTickCount;

    if (GameInit(hInstance))
    {
        if (!GameLib()->Initialize(nCmdShow))
        {
            return FALSE;
        }

        // Main message loop:
        while (TRUE)
        {
            try
            {
                if (PeekMessage(&msg, NULL, 0, 0, PM_REMOVE))
                {
                    if( msg.message == WM_QUIT )
                        break;

                    TranslateMessage(&msg);
                    DispatchMessage(&msg);
                }
            }
            else
            {
                if (!GameLib()->GetHibernate())
                {
                    iTickCount = GetTickCount();
                    if (iTickCount > iNewTime)
                    {
                        iNewTime = iTickCount + GameLib()->GetFrameRate();

                        GameEvent();
                    }
                }
            }
        }
        _except(EXCEPTION_EXECUTE_HANDLER)
        {
            MessageBox(NULL, _T("Exception error!"), _T("Error"), MB_OK | MB_ICONERROR);
        }
    }
}

```



```

        return msg.wParam;
    }

    GameEnd();
    return 1;
}

////////////////////////////////////
// MsgBox
////////////////////////////////////
void MsgBox(LPTSTR szMessage)
{
    MessageBox(GetForegroundWindow(), szMessage, GameLib()->szTitle, MB_OK |
    MB_ICONINFORMATION);
}

```

```

////////////////////////////////////
// Pengaturcaraan Permainan Pocket PC
// Kod Sumber CAnimSprite
////////////////////////////////////

#include "stdafx.h"
#include "CSprite.h"
#include "CAnimSprite.h"

```

```

////////////////////////////////////
// CAnimSprite Constructor
////////////////////////////////////
CAnimSprite::CAnimSprite(HDC hdc): CSprite(hdc)
{
    iCurFrame = 0;
    iTotalFrames = 0;

    for (int n = 0; n < NUM_FRAMES; n++)
        sprites[n] = new CBitmap(hdc);
}

```

```

////////////////////////////////////
// CAnimSprite Destructor
////////////////////////////////////
CAnimSprite::~CAnimSprite()
{
    for (int n = 0; n < NUM_FRAMES; n++)
        delete sprites[n];
}

```

```

////////////////////////////////////
// CAnimSprite::SetImageSize
////////////////////////////////////
void CAnimSprite::SetImageSize(int iImgWidth, int iImgHeight)
{
    iWidth = iImgWidth;
    iHeight = iImgHeight;
}

```

```

////////////////////////////////////
// CAnimSprite::LoadTile
////////////////////////////////////

```

```
BOOL CAnimSprite::LoadTile(int iTileNum, LPTSTR lpFilename, int iWidth, int iHeight)
```

```
{
    CBitmap *source;
    int iStartX, iStartY;

    if (iTileNum < 0 || iTileNum > NUM_FRAMES)
        return FALSE;

    if (iWidth < 1 || iWidth > 240 || iHeight < 1 || iHeight > 320)
        return FALSE;

    source = new CBitmap(GetScreenDC());
    if (!source->Load(lpFilename))
    {
        delete source;
        return FALSE;
    }

    iStartX = iTileNum * (iWidth + 1);
    iStartY = 0;

    if (!sprites[iTileNum]->Create(iWidth, iHeight))
    {
        delete source;
        return FALSE;
    }

    BitBlt(sprites[iTileNum]->GetSourceDC(),
        0, 0, iWidth, iHeight,
        source->GetSourceDC(),
        iStartX, iStartY, SRCCOPY);

    iTotFrames++;
    SetFrame(iTileNum);
    SetImageSize(iWidth, iHeight);
    delete source;

    return TRUE;
}
```

```
////////////////////////////////////
// CAnimSprite::LoadAnimSeq
////////////////////////////////////
```

```
BOOL CAnimSprite::LoadAnimSeq(int iStartCol, int iNumFrames, int iRow, LPTSTR lpFilename, int iWidth, int iHeight)
```

```
{
    CBitmap *source;
    int iStartX, iStartY, iFrame;

    if (iStartCol < 0 || iNumFrames > NUM_FRAMES)
        return FALSE;

    if (iWidth < 1 || iWidth > 240 || iHeight < 1 || iHeight > 320)
        return FALSE;

    source = new CBitmap(GetScreenDC());
    if (!source->Load(lpFilename))
    {
        delete source;
        return FALSE;
    }
}
```



```

}

for (iFrame = 0; iFrame < iNumFrames; iFrame++)
{
    if (!sprites[iFrame]->Create(iWidth, iHeight))
    {
        delete source;
        return FALSE;
    }

    iStartX = iStartCol * (iWidth + 1) + iFrame * (iWidth + 1);
    iStartY = iRow * (iHeight + 1);

    BitBlt(sprites[iFrame]->GetSourceDC(),
        0, 0, iWidth, iHeight,
        source->GetSourceDC(),
        iStartX, iStartY, SRCCOPY);

    iTotalFrames++;
}

SetFrame(iStartCol);
SetImageSize(iWidth, iHeight);
delete source;

return TRUE;
}

```

```

////////////////////////////////////
// CAnimSprite::BitBlit
////////////////////////////////////
BOOL CAnimSprite::BitBlit()
{
    return BitBlt(GetScreenDC(), GetX(), GetY(),
        ImageWidth(), ImageHeight(),
        sprites[iCurFrame]->GetSourceDC(), 0, 0, SRCCOPY);
}

```

```

////////////////////////////////////
// CAnimSprite::StretchBlit
////////////////////////////////////
BOOL CAnimSprite::StretchBlit(int dx, int dy)
{
    return StretchBlt(GetScreenDC(),
        GetX(), GetY(), dx, dy,
        sprites[iCurFrame]->GetSourceDC(),
        0, 0, ImageWidth(), ImageHeight(), SRCCOPY);
}

```

```

////////////////////////////////////
// CAnimSprite::TransBlit
////////////////////////////////////
BOOL CAnimSprite::TransBlit(COLORREF clrTrans)
{
    return TransparentImage(GetScreenDC(),
        GetX(), GetY(),
        ImageWidth(), ImageHeight(),
        sprites[iCurFrame]->GetSourceDC(), 0, 0,
        ImageWidth(), ImageHeight(), clrTrans);
}

```

```

////////////////////////////////////
// CAnimSprite::SetFrame
////////////////////////////////////
void CAnimSprite::SetFrame(int iFrame)
{
    if (iFrame >= 0 && iFrame < iTotFrames)
        iCurFrame = iFrame;
}

////////////////////////////////////
// CAnimSprite::NextFrame
////////////////////////////////////
void CAnimSprite::NextFrame()
{
    if (iCurFrame < iTotFrames - 1)
        iCurFrame++;
    else
        iCurFrame = 0;
}

////////////////////////////////////
// CAnimSprite::PrevFrame
////////////////////////////////////
void CAnimSprite::PrevFrame()
{
    if (iCurFrame > 0)
        iCurFrame--;
    else
        iCurFrame = iTotFrames - 1;
}

```

```

////////////////////////////////////
// Pengaturcaraan Permainan Pocket PC
// Kod Sumber CBitmap
////////////////////////////////////

#include "stdafx.h"
#include "CBitmap.h"

```

```

////////////////////////////////////
// CBitmap Constructor
////////////////////////////////////

```

```

CBitmap::CBitmap(HDC hdc)
{
    hScreenDC = hdc;
}

```

```

////////////////////////////////////
// CBitmap Destructor
////////////////////////////////////
CBitmap::~CBitmap()
{
    if (lpDIB != NULL)
        free(lpDIB);

```

```

    if (hSourceDC != NULL)
    {

```



```

        SelectObject(hSourceDC, hOldBitmap);
        DeleteObject(hOldBitmap);
        DeleteDC(hSourceDC);
    }
}

////////////////////////////////////
// CBitmap::Create
////////////////////////////////////
BOOL CBitmap::Create(int Width, int Height)
{
    HBITMAP hSourceBitmap;

    hSourceDC = CreateCompatibleDC(GetScreenDC());
    if (hSourceDC == NULL)
        return FALSE;

    hSourceBitmap = CreateCompatibleBitmap(GetScreenDC(), Width, Height);
    if (hSourceBitmap == NULL)
        return FALSE;

    hOldBitmap = SelectBitmap(hSourceDC, hSourceBitmap);
    if (hOldBitmap == NULL)
        return FALSE;

    DeleteObject(hSourceBitmap);
    return TRUE;
}

////////////////////////////////////
// CBitmap::Load
////////////////////////////////////
BOOL CBitmap::Load(LPTSTR lpFilename)
{
    LPBYTE lpSourceBits;
    HBITMAP hSourceBitmap;
    DWORD dwSourceBitsSize;
    LPBITMAPINFO lpSrcDIB;

    SetFilename(lpFilename);
    if (!LoadDIB(lpFilename))
        return FALSE;

    lpSrcDIB = (LPBITMAPINFO)lpDIB;

    hSourceBitmap = CreateDIBSection(GetScreenDC(), lpSrcDIB, DIB_RGB_COLORS,
        (void **)&lpSourceBits, NULL, 0);

    hSourceDC = CreateCompatibleDC(GetScreenDC());
    dwSourceBitsSize = lpSrcDIB->bmiHeader.biHeight *
        BytesPerLine(&(lpSrcDIB->bmiHeader));
    memcpy(lpSourceBits, BitmapDataIndex((LPSTR)lpSrcDIB), dwSourceBitsSize);

    SelectObject(hSourceDC, hSourceBitmap);

    //simpan header dalam bmBitmap
    GetObject(hSourceBitmap, sizeof(BITMAP), &bmBitmap);

    DeleteObject(hSourceBitmap);

```

```

    return TRUE;
}

////////////////////////////////////
// CBitmap::LoadDIB
////////////////////////////////////
BOOL CBitmap::LoadDIB(LPCTSTR szFileName)
{
    HANDLE hFile;
    BITMAPFILEHEADER bfh;
    LPBYTE lpTemp = NULL;
    WORD wPaletteSize = 0;
    DWORD dwBitsSize = 0;
    BOOL bRet;

    //buka fail
    hFile = CreateFile(szFileName, GENERIC_READ, FILE_SHARE_READ, NULL,
        OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL, NULL);
    if (hFile == INVALID_HANDLE_VALUE)
    {
        return FALSE;
    }

    //baca header
    bRet = ReadFile(hFile, &bfh, sizeof(BITMAPFILEHEADER), &dwImageSize, NULL);
    if ((bRet == FALSE) || (dwImageSize != sizeof(BITMAPFILEHEADER)))
    {
        CloseHandle(hFile);
        return FALSE;
    }

    //adakah fail bitmap berguna?
    if ((bfh.bfType != 0x4d42) || (bfh.bfReserved1 != 0) || (bfh.bfReserved2 != 0))
    {
        CloseHandle(hFile);
        return FALSE;
    }

    //cari memori untuk data bitmap
    lpDIB = (LPBYTE) malloc(sizeof(BITMAPINFO));
    if (lpDIB == NULL)
    {
        CloseHandle(hFile);
        return FALSE;
    }

    //baca BITMAPINFOHEADER
    bRet = ReadFile(hFile, lpDIB, sizeof(BITMAPINFOHEADER), &dwImageSize, NULL);
    if ((bRet == FALSE) || (dwImageSize != sizeof(BITMAPINFOHEADER)))
    {
        CloseHandle(hFile);
        free(lpDIB);
        return FALSE;
    }

    if (((LPBITMAPINFOHEADER)lpDIB)->biSize != sizeof(BITMAPINFOHEADER))
    {
        CloseHandle(hFile);
        free(lpDIB);
        return FALSE;
    }
}

```



```

    }

    wPaletteSize = PaletteSize((LPSTR)lpDIB);
    dwBitsSize = ImageHeight() * BytesPerLine((LPBITMAPINFOHEADER) lpDIB);

    //cari DIB buffer
    lpTemp = (LPBYTE) realloc(lpDIB, sizeof(BITMAPINFOHEADER) + wPaletteSize +
dwBitsSize);
    if (lpTemp == NULL)
    {
        CloseHandle(hFile);
        free(lpDIB);
        return FALSE;
    }

    lpDIB = lpTemp;

    //baca pallete jika ada
    if (wPaletteSize > 0)
    {
        bRet = ReadFile(hFile, ((LPBITMAPINFO)lpDIB)->bmiColors, wPaletteSize,
&dwImageSize, NULL);
        if ((bRet == FALSE) || (dwImageSize != wPaletteSize))
        {
            CloseHandle(hFile);
            free(lpDIB);
            return FALSE;
        }
    }

    if (bfh.bfOffBits > 0)
    {
        if (SetFilePointer(hFile, bfh.bfOffBits, NULL, FILE_BEGIN) == 0xffffffff)
        {
            CloseHandle(hFile);
            free(lpDIB);
            return FALSE;
        }
    }

    //baca data imej
    bRet = ReadFile(hFile, BitmapDataIndex((LPSTR)lpDIB), dwBitsSize, &dwImageSize,
NULL);
    if ((bRet == FALSE) || (dwImageSize != dwBitsSize))
    {
        CloseHandle(hFile);
        free(lpDIB);
        return FALSE;
    }

    //tutup fail
    CloseHandle(hFile);
    return TRUE;
}

////////////////////////////////////
// CBitmap::BitBlit
////////////////////////////////////
BOOL CBitmap::BitBlit(int x, int y)

```

```

{
    return BitBlt(GetScreenDC(), x, y, ImageWidth(), ImageHeight(), hSourceDC, 0, 0,
SRCCOPY);
}

////////////////////////////////////
// CBitmap::BitBlit (Overloaded)
////////////////////////////////////
BOOL CBitmap::BitBlit(HDC hdc, int x, int y)
{
    return BitBlt(hdc, x, y, ImageWidth(), ImageHeight(), hSourceDC, 0, 0, SRCCOPY);
}

////////////////////////////////////
// CBitmap::TransBlit
////////////////////////////////////
BOOL CBitmap::TransBlit(int x, int y, COLORREF clrTrans)
{
    return TransparentImage(GetScreenDC(), x, y, ImageWidth(), ImageHeight(), hSourceDC, 0, 0,
        ImageWidth(), ImageHeight(), clrTrans);
}

////////////////////////////////////
// CBitmap::StretchBlit
////////////////////////////////////
BOOL CBitmap::StretchBlit(int x, int y, int dx, int dy)
{
    return StretchBlt(GetScreenDC(), x, y, dx, dy, hSourceDC, 0, 0, ImageWidth(),
ImageHeight(), SRCCOPY);
}

////////////////////////////////////
// CBitmap::BitmapDataIndex
////////////////////////////////////
LPSTR CBitmap::BitmapDataIndex(LPSTR lpbi)
{
    return (lpbi + *(LPDWORD)lpbi + PaletteSize(lpbi));
}

////////////////////////////////////
// CBitmap::PaletteSize
////////////////////////////////////
int CBitmap::PaletteSize(LPSTR lpbi)
{
    int iNumColors;
    DWORD dwClrUsed;

    dwClrUsed = ((LPBITMAPINFOHEADER) lpbi)->biClrUsed;
    if (dwClrUsed)
    {
        iNumColors = (WORD)dwClrUsed;
    }
    else
    {
        //untuk kurang dari 8-bit
        switch (BitCount())
        {

```



```

        case 1:
            iNumColors = 2;
            break;

        case 2:
            iNumColors = 4;
            break;

        case 4:
            iNumColors = 16;
            break;

        case 8:
            iNumColors = 256;
            break;

        default:
            iNumColors = 0;
    }
}

return (iNumColors * sizeof(RGBQUAD));
}

////////////////////////////////////
// CBitmap::BytesPerLine
////////////////////////////////////
int CBitmap::BytesPerLine(LPBITMAPINFOHEADER lpBMIH)
{
    return (int)(((ImageWidth() * NumPlanes() * BitCount() + 31) / 32) * 4);
}

////////////////////////////////////
// CBitmap::BitCount
////////////////////////////////////
int CBitmap::BitCount()
{
    return (int)((LPBITMAPINFOHEADER) lpDIB->biBitCount;
}

////////////////////////////////////
// CBitmap::ImageWidth
////////////////////////////////////
int CBitmap::ImageWidth()
{
    return (int)((LPBITMAPINFOHEADER) lpDIB->biWidth;
}

////////////////////////////////////
// CBitmap::ImageHeight
////////////////////////////////////
int CBitmap::ImageHeight()
{
    return (int)((LPBITMAPINFOHEADER) lpDIB->biHeight;
}

////////////////////////////////////
// CBitmap::NumPlanes
////////////////////////////////////
int CBitmap::NumPlanes()

```

```
{
    return (int)((LPBITMAPINFOHEADER) lpDIB)->biPlanes;
}
```

```
////////////////////////////////////
// Pengaturcaraan Permainan Pocket PC
// Kod Sumber CSprite
////////////////////////////////////
```

```
#include "stdafx.h"
#include "CBitmap.h"
#include "CSprite.h"
```

```
////////////////////////////////////
// CSprite Constructor
////////////////////////////////////
CSprite::CSprite():CBitmap()
{
    Init();
}
```

```
////////////////////////////////////
// CSprite Constructor (Overloaded)
////////////////////////////////////
CSprite::CSprite(HDC hdc):CBitmap(hdc)
{
    Init();
}
```

```
void CSprite::Init()
```

```
{
    X_Loc = 0;
    Y_Loc = 0;
    X_Dir = 1;
    Y_Dir = 1;
    X_Spd = 1;
    Y_Spd = 1;

    bUnderSaved = FALSE;
    bAlive = FALSE;

    DestRect = new RECT;
    Rect1 = new RECT;
    Rect2 = new RECT;
}
```

```
////////////////////////////////////
// CSprite Destructor
////////////////////////////////////
CSprite::~CSprite()
{
    DeleteDC(hdcUnder);

    delete DestRect;
    delete Rect1;
    delete Rect2;
}
```



```

////////////////////////////////////
// CSprite::SaveUnder
////////////////////////////////////
BOOL CSprite::SaveUnder(HDC hdc)
{
    if (!bUnderSaved)
    {
        hUnder = CreateCompatibleBitmap(hdc, ImageWidth(), ImageHeight());
        if (hUnder == NULL)
            return FALSE;

        hdcUnder = CreateCompatibleDC(hdc);
        if (hdcUnder == NULL)
            return FALSE;

        SelectObject(hdcUnder, hUnder);
        DeleteObject(hUnder);
        bUnderSaved = TRUE;
    }

    if (hdcUnder != 0 && hdc != 0)
        BitBlt(hdcUnder, 0, 0, ImageWidth(), ImageHeight(), hdc, GetX(), GetY(), SRCCOPY);

    return TRUE;
}

////////////////////////////////////
// CSprite::RestoreUnder
////////////////////////////////////
BOOL CSprite::RestoreUnder(HDC hdc)
{
    if (hdcUnder != 0 && hdc != 0)
        BitBlt(hdc, GetX(), GetY(), ImageWidth(), ImageHeight(), hdcUnder, 0, 0, SRCCOPY);
    else
        return FALSE;

    return TRUE;
}

////////////////////////////////////
// CSprite::TransBlit
////////////////////////////////////
BOOL CSprite::TransBlit(COLORREF clrTrans)
{
    return CBitmap::TransBlit(GetX(), GetY(), clrTrans);
}

////////////////////////////////////
// CSprite::BitBlit
////////////////////////////////////
BOOL CSprite::BitBlit()
{
    return CBitmap::BitBlit(GetX(), GetY());
}

////////////////////////////////////
// CSprite::BitBlit (Overloaded)
////////////////////////////////////
BOOL CSprite::BitBlit(HDC hdc)

```

```

{
    return CBitmap::BitBlit(hdc, GetX(), GetY());
}

////////////////////////////////////
// CSprite::StretchBlit
////////////////////////////////////
BOOL CSprite::StretchBlit(int dx, int dy)
{
    return CBitmap::StretchBlit(GetX(), GetY(), dx, dy);
}

////////////////////////////////////
// CSprite::Intersected
////////////////////////////////////
BOOL CSprite::Intersected(CSprite *Sprite, POINT &pt)
{
    RECT rt;

    if (Intersected(Sprite, rt))
    {
        pt.x = (rt.left + rt.right) / 2;
        pt.y = (rt.top + rt.bottom) / 2;
        return TRUE;
    }

    return FALSE;
}

////////////////////////////////////
// CSprite::Intersected
////////////////////////////////////
BOOL CSprite::Intersected(CSprite *Sprite, RECT &rt)
{
    int iShrinkX, iShrinkY;

    iShrinkX = ImageWidth() >> 2;
    iShrinkY = ImageHeight() >> 2;

    Rect1->left = GetX() + iShrinkX;
    Rect1->top = GetY() + iShrinkY;
    Rect1->right = GetX() + ImageWidth() - iShrinkX;
    Rect1->bottom = GetY() + ImageHeight() - iShrinkY;

    iShrinkX = Sprite->ImageWidth() >> 2;
    iShrinkY = Sprite->ImageHeight() >> 2;

    Rect2->left = Sprite->GetX() + iShrinkX;
    Rect2->top = Sprite->GetY() + iShrinkY;
    Rect2->right = Sprite->GetX() + Sprite->ImageWidth() - iShrinkX;
    Rect2->bottom = Sprite->GetY() + Sprite->ImageHeight() - iShrinkY;

    if (IntersectRect(DestRect, Rect1, Rect2))
    {
        rt.left = DestRect->left;
        rt.right = DestRect->right;
        rt.top = DestRect->top;
        rt.bottom = DestRect->bottom;
        return TRUE;
    }
}

```



```

    return FALSE;
}

```

```

////////////////////////////////////
// Pengaturcaraan Permainan Pocket PC
// Header CGameLibrary
////////////////////////////////////

```

```

#pragma once

```

```

#include "stdafx.h"

```

```

//fungsi windows utama

```

```

int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPTSTR lpCmdLine,
int nCmdShow);
LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM
lParam);

```

```

//fungsi2 GameLibrary

```

```

extern "C"

```

```

{
    BOOL GameInit(HINSTANCE hInst);
    void GameStart(HWND hWnd);
    void GameEnd();
    void GameActivate(HWND hWnd);
    void GameEvent();
    void GamePaint(HWND hWnd);
    void StylusDown(int x, int y);
    void StylusMove(int x, int y);
    void StylusUp(int x, int y);
    void ButtonPress(int iButtonID, POINT pt);
    void ButtonRelease(int iButtonID, POINT pt);
}

```

```

struct COLOR

```

```

{
    BYTE Red;
    BYTE Green;
    BYTE Blue;
};

```

```

//fungsi sokongan

```

```

void MsgBox(LPTSTR szMessage);

```

```

////////////////////////////////////
// CGameLibrary
////////////////////////////////////

```

```

class CGameLibrary

```

```

{

```

```

protected:

```

```

    //pointer ke objek CGameLibrary

```

```

    static CGameLibrary *pGameLib;

```

```

    TCHAR szWindowClass[40];

```

```

    TCHAR szTitle[40];

```

```

    BOOL bHibernate;

```

```

    BOOL bFullscreen;

```

```

    WORD wProgramIcon;

```

```

int iFrameRate;

//GDI-specific
HINSTANCE hInstance;
HWND hWindow;

//GAPI-specific
    unsigned char *GAPIVidMem;
BOOL bGAPIDisplay;
GXDisplayProperties gxDisplay;

GXKeyList gxKeys;

public:
    CGameLibrary(HINSTANCE hInst, LPTSTR szNewWindowClass);
    virtual ~CGameLibrary();

    BOOL Initialize(int nCmdShow);
    LRESULT EventHandler(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam);
    void Error(LPTSTR szError);
    void Shutdown();
    int ScreenWidth() { return GetSystemMetrics(SM_CXSCREEN); };
    int ScreenHeight() { return GetSystemMetrics(SM_CYSCREEN); };

    void PrintText(HDC hdc, LPCTSTR strText, int x, int y, COLORREF color, int iBkMode);
    void PrintTextXY(HDC hdc, LPCTSTR strText, int x, int y, COLORREF color, int iBkMode);
    void FatalError(LPCTSTR lpStr);
    LPWSTR GetPath(TCHAR *filename = _T(""));
    void Delay(int ms);
    BOOL PointInside(POINT pt, int left, int top, int right, int bottom);
    BOOL PointInside(POINT pt, RECT rt);

    HINSTANCE GetInstance() { return hInstance; };
    void SetInstance(HINSTANCE hInst) { hInstance = hInst; };

    HWND GetWindow() { return hWindow; };
    void SetWindow(HWND hWnd) { hWindow = hWnd; };

    LPTSTR GetTitle() { return szTitle; };
    void SetTitle(LPTSTR szNewTitle) { wcsncpy(szTitle, szNewTitle); };

    BOOL GetHibernate() { return bHibernate; };
    void SetHibernate(BOOL bSet) { bHibernate = bSet; };

    BOOL GetFullscreen() { return bFullscreen; };
    void SetFullscreen(BOOL bSet) { bFullscreen = bSet; };

    WORD GetIcon() { return wProgramIcon; };
    void SetIcon(WORD wIcon) { wProgramIcon = wIcon; };

    int GetFrameRate() { return iFrameRate; };
    void SetFrameRate(int iFrames) { iFrameRate = 1000/iFrames; };

    BOOL G_Enabled() { return bGAPIDisplay; };
    void G_SetDisplay(BOOL bSet) { bGAPIDisplay = bSet; };
    void G_BeginDraw();
    void G_EndDraw();
    int GetXPitch() { return gxDisplay.cbxPitch; };
    int GetYPitch() { return gxDisplay.cbyPitch; };
    int GetBitsPerPixel() { return gxDisplay.cbPP; };

```



```

BOOL IsScreenFormat565() { return (gxDisplay.iffFormat | kfDirect565); };
unsigned char *G_GetVidMem() { return GAPIVidMem; };
BOOL G_ClearScreen(COLOR color);
int G_DrawPixel16(unsigned char *VidMem, int X, int Y, COLOR color);
void G_DrawSolidRect(unsigned char *VidMem, int iLeft, int iTop, int iRight, int iBottom, COLOR
color);

static CGameLibrary *GetObj(void) { return pGameLib; };

friend void MsgBox(LPTSTR szMessage);

};

inline CGameLibrary *GameLib() { return CGameLibrary::GetObj(); };

```

```

////////////////////////////////////
// Pengaturcaraan Permainan Pocket PC
// Header CAnimSprite
////////////////////////////////////

```

```
#pragma once
```

```
#include "stdafx.h"
#include "CSprite.h"

```

```
#define NUM_FRAMES 16
```

```
class CAnimSprite: public CSprite
{
private:

```

```

//CBitmap *source;
CBitmap *sprites[NUM_FRAMES];
int iCurFrame, iTotalFrames;
int iWidth, iHeight;

```

```

HBITMAP hUnder;
HDC hdcUnder;

```

```
public:
```

```

CAnimSprite(HDC hdc);
virtual ~CAnimSprite();
BOOL LoadTile(int iTileNum, LPTSTR lpFilename, int iWidth, int iHeight);
BOOL LoadAnimSeq(int iStartCol, int iNumFrames, int iRow, LPTSTR lpFilename, int iWidth, int
iHeight);

```

```

virtual BOOL BitBlit();
virtual BOOL StretchBlit(int dx, int dy);
virtual BOOL TransBlit(COLORREF clrTrans);

```

```

int ImageWidth() { return iWidth; };
int ImageHeight() { return iHeight; };
void SetImageSize(int iImgWidth, int iImgHeight);

```

```

void SetFrame(int iFrame);
int GetFrame() { return iCurFrame; };
void NextFrame();
void PrevFrame();

```

```

void SetTotalFrames(int iNew) { iTotalFrames = iNew; };
int GetTotalFrames() { return iTotalFrames; };

};

```

```

////////////////////////////////////
// Pengaturcaraan Permainan Pocket PC
// Header CBitmap
////////////////////////////////////

```

```

#pragma once

```

```

class CBitmap
{

```

```

protected:

```

```

    HDC hScreenDC;
    LPBYTE lpDIB;
    LPBYTE lpSourceBits;
    LPBITMAPINFO lpbmi;
    DWORD dwImageSize;
    BITMAP bmBitmap;
    LPTSTR lpBitmapFN;
    HDC hSourceDC;
    HBITMAP hOldBitmap;

```

```

    BOOL LoadDIB(LPCTSTR szFileName);
    LPSTR BitmapDataIndex(LPSTR lpbi);
    int PaletteSize(LPSTR lpbi);
    int BytesPerLine(LPBITMAPINFOHEADER lpBMIH);

```

```

public:

```

```

    CBitmap() { };
    CBitmap(HDC hdc);
    virtual ~CBitmap();
    virtual HDC GetSourceDC() { return hSourceDC; };
    virtual BOOL BitBlit(HDC hdc, int x, int y);
    virtual BOOL BitBlit(int x, int y);
    virtual BOOL StretchBlit(int x, int y, int dx, int dy);
    virtual BOOL TransBlit(int x, int y, COLORREF clrTrans);

```

```

    LPTSTR GetFilename() { return lpBitmapFN; };
    void SetFilename(LPTSTR lpFN) { lpBitmapFN = lpFN; };
    HDC GetScreenDC() { return hScreenDC; };
    void SetScreenDC(HDC hNew) { hScreenDC = hNew; };

```

```

    BOOL Create(int Width, int Height);
    BOOL Load(LPTSTR lpFilename);
    int BitCount();
    virtual int ImageWidth();
    virtual int ImageHeight();
    int NumPlanes();

```

```

};

```

```

////////////////////////////////////
// Pengaturcaraan Permainan Pocket PC
// Header CSprite

```



```
////////////////////////////////////
```

```
#pragma once
```

```
#include "stdafx.h"
```

```
#include "CBitmap.h"
```

```
class CSprite : public CBitmap
```

```
{
private:
    int X_Loc, Y_Loc;
    int X_Dir, Y_Dir;
    int X_Spd, Y_Spd;

    HBITMAP hUnder;
    HDC hdcUnder;
    BOOL bUnderSaved;
    BOOL bAlive;
    LPRECT DestRect;
    RECT *Rect1;
    RECT *Rect2;

    void Init();

public:
    CSprite();
    CSprite(HDC hdc);
    virtual ~CSprite();
    BOOL SaveUnder(HDC hdc);
    BOOL RestoreUnder(HDC hdc);
    virtual BOOL TransBlit(COLORREF clrTrans);
    virtual BOOL BitBlit();
    virtual BOOL BitBlit(HDC hdc);
    virtual BOOL StretchBlit(int dx, int dy);
    virtual int ImageWidth() { return CBitmap::ImageWidth(); };
    virtual int ImageHeight() { return CBitmap::ImageHeight(); };
    virtual BOOL Intersected(CSprite *Sprite, POINT &pt);
    virtual BOOL Intersected(CSprite *Sprite, RECT &rt);

    BOOL IsAlive() { return bAlive; };
    void SetAlive(BOOL bNew) { bAlive = bNew; };
    int GetX() { return X_Loc; };
    void SetX(int vNew) { X_Loc = vNew; };
    void IncX(int vNew) { X_Loc += vNew; };
    int GetY() { return Y_Loc; };
    void SetY(int vNew) { Y_Loc = vNew; };
    void IncY(int vNew) { Y_Loc += vNew; };
    int GetXDir() { return X_Dir; };
    void SetXDir(int vNew) { X_Dir = vNew; };
    int GetYDir() { return Y_Dir; };
    void SetYDir(int vNew) { Y_Dir = vNew; };
    int GetXSpeed() { return X_Spd; };
    void SetXSpeed(int vNew) { X_Spd = vNew; };
    int GetYSpeed() { return Y_Spd; };
    void SetYSpeed(int vNew) { Y_Spd = vNew; };
};
```

```
//////////////////////////////////
// StaAfx.h
//////////////////////////////////

#if !defined(AFX_STDAFX_H__65FE072C_4CD9_4541_A750_426999E4B77F__INCLUDED_)
#define AFX_STDAFX_H__65FE072C_4CD9_4541_A750_426999E4B77F__INCLUDED_

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000

#define WIN32_LEAN_AND_MEAN

#include <windows.h>
#include <windowsx.h>
#include <aygshell.h>
#include <commctrl.h>
#include <shellapi.h>
#include <sipapi.h>
#include <stdlib.h>
#include <winbase.h>
#include <windef.h>
#include <wingdi.h>
#include <mmsystem.h>
#include <gx.h>

#include <af_irda.h>

#endif //
!defined(AFX_STDAFX_H__65FE072C_4CD9_4541_A750_426999E4B77F__INCLUDED_)
```


MANUAL UNTUK PENGGUNA

MANUAL PENGGUNA

University

10

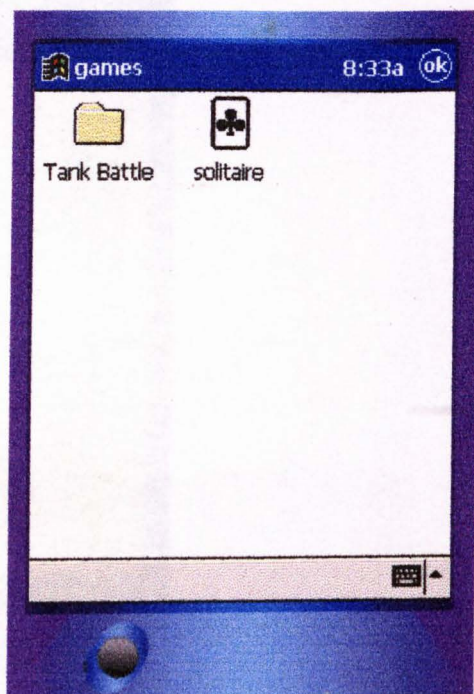
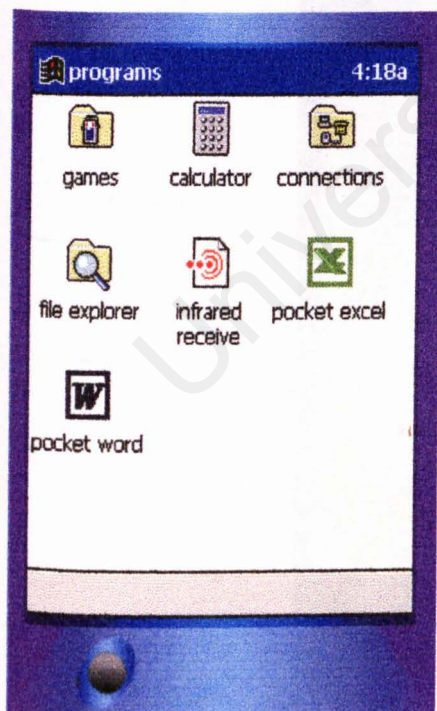
University of Technology, Sydney

University of Technology, Sydney

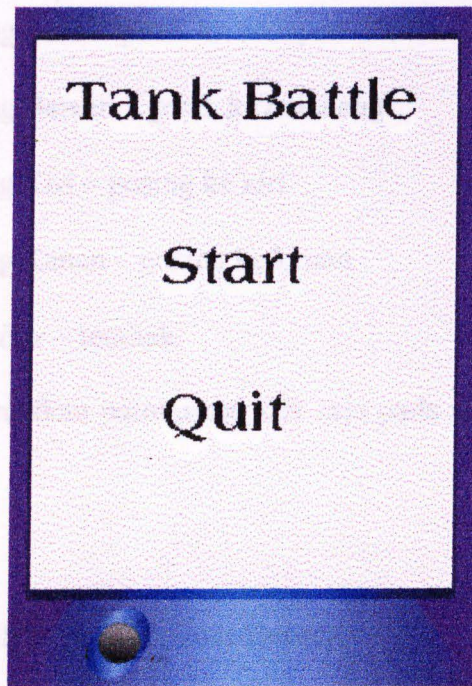
MANUAL UNTUK PENGGUNA



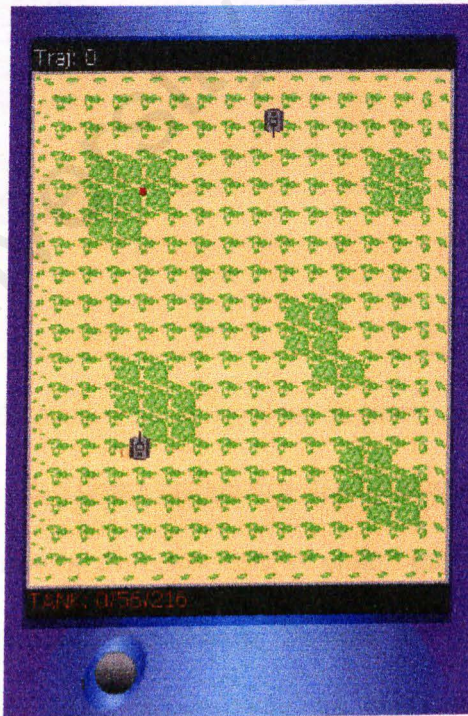
Langkah 1 : Pengguna hendaklah menekan butang *Start* untuk memilih aplikasi dan kemudian pilih *Programs*.



Langkah 2 : Dari antaramuka *Programs*, pilih *Games*. Terdapat satu direktori bernama *Tank Battle*. Klik pada direktori tersebut.



Langkah 3 : Terdapat beberapa fail bitmap dan satu fail aplikasi bernama Project01. Klik pada fail aplikasi tersebut dan akan keluar paparan menu utama seperti di atas. Jika hendak bermain, klik *Start*. Kalau tidak, klik *Quit*.



Langkah 4 : Terdapat 2 buah kereta kebal di mana yang bawah dikawal pengguna dan yang atas dikawal komputer. Untuk menggerakkan kereta kebal memerlukan butang Atas, Bawah, Kiri, Kanan dan *SpaceBar*.

Rujukan

- [1] Redmond-Pyle David, Moore Alan, *Graphical User Interface Design and Evaluation*, Prentice Hall, 1995
- [2] Analisa Intelektual Komputer dan Manusia
[3] Nita Fidiyati Marlin, *System Mathematik Logika Industri*, UM, 2001
- [4] Steve McPherson, *How To Do Everything With Your Pocket PC and Handheld PC*, Osborne/McGraw-Hill, 1999

RUJUKAN

- [5] ...
- [6] ...
- [7] ...
- [8] ...
- [9] ...
- [10] ...
- [11] ...
- [12] ...
- [13] ...
- [14] ...
- [15] ...
- [16] ...
- [17] ...
- [18] ...
- [19] ...
- [20] ...
- [21] ...
- [22] ...
- [23] ...
- [24] ...
- [25] ...
- [26] ...
- [27] ...
- [28] ...
- [29] ...
- [30] ...
- [31] ...
- [32] ...
- [33] ...
- [34] ...
- [35] ...
- [36] ...
- [37] ...
- [38] ...
- [39] ...
- [40] ...
- [41] ...
- [42] ...
- [43] ...
- [44] ...
- [45] ...
- [46] ...
- [47] ...
- [48] ...
- [49] ...
- [50] ...
- [51] ...
- [52] ...
- [53] ...
- [54] ...
- [55] ...
- [56] ...
- [57] ...
- [58] ...
- [59] ...
- [60] ...
- [61] ...
- [62] ...
- [63] ...
- [64] ...
- [65] ...
- [66] ...
- [67] ...
- [68] ...
- [69] ...
- [70] ...
- [71] ...
- [72] ...
- [73] ...
- [74] ...
- [75] ...
- [76] ...
- [77] ...
- [78] ...
- [79] ...
- [80] ...
- [81] ...
- [82] ...
- [83] ...
- [84] ...
- [85] ...
- [86] ...
- [87] ...
- [88] ...
- [89] ...
- [90] ...
- [91] ...
- [92] ...
- [93] ...
- [94] ...
- [95] ...
- [96] ...
- [97] ...
- [98] ...
- [99] ...
- [100] ...

Rujukan

- [1] Redmond-Pyle David, Moore Alan, ***Graphical User Interface Design and Evaluation***, Prentice Hall, 1995
- [2] Analisa Interaksi Komputer dan Manusia
www.acm.org/sigchi
- [3] Nor Edelina Mazlin, ***Sistem Maklumat Latihan Industri***, UM, 2001
- [4] Frank McPherson, ***How To Do Everything With Your Pocket PC and Handheld PC***, Osborne McGraw-Hill, 1999
- [5] Jonathan S. Harbour, ***Pocket PC Game Programming : Using The Windows® CE Game API***, Prima Publishing, 2000
- [6] *Here Comes GAPI!*
<http://www.microsoft.com/mobile/developer/default.asp>
- [7] Todd Baron, ***Multiplayer Game Programming***, Prima Publishing, 1999
- [8] *Catcha Malaysia-Wireless*
www.catcha.com.my/wireless
- [9] Shari Lawrence Pfleeger, ***Software Engineering : Theory And Practice (2nd Edition)***, Prentice Hall, 2001
- [10] Jesse Liberty, ***C++ : An Introduction to Programming***, Que Education and Training, 1999
- [11] Roger T. Sevens, ***Advance Graphics Programming in C and C++***, M&T Publishing, Inc. , 2000
- [12] Andr  LaMothe, ***Teach Yourself Game Programming***, Sam's Publishing, 1999

RUMUSAN

Projek ini dibangunkan di bawah subjek latihan ilmiah tahap akhir II (WXES 3182). Tajuk bagi projek ini ialah Permainan Untuk Pocket PC dan diselia oleh Cik Siti Hafizah. Moderator bagi projek ini ialah Puan Raja Noor Ainun.

Untuk menjayakan projek ini, pelbagai usaha telah dicurahkan untuk mencari maklumat, membuat rujukan dan membangunkan sistem. Segala maklumat dicari di pelbagai tempat seperti perpustakaan, temuramah, majalah dan lain-lain lagi. Setelah pelbagai analisa dibuat, pengaturcaraan yang akan digunakan untuk membuat permainan beranimasi ialah Microsoft Embedded Visual Tools 3.0 dengan Windows CE sebagai pelantar.

Projek ini dibangunkan melalui beberapa fasa iaitu analisis, rekabentuk, implementasi, pengujian, penilaian dan dokumentasi. Permainan yang dibangunkan mestilah memenuhi ciri-ciri Pocket PC dan saiz permainan hendaklah sekecil yang boleh tetapi mempunyai cara bermain yang menarik. Selain itu, permainan tersebut hendaklah mempunyai animasi yang menarik dan mudah difahami pengguna.

Faedah yang dapat diperolehi di sini ialah kita dapat mengikuti perkembangan semasa dalam evolusi perkembangan perkakasan PDA (Personal Digital Assistant) dan evolusi permainan beranimasi. Memandangkan teknologi sains komputer dan IT yang semakin berkembang maju, permainan beranimasi juga tidak ketinggalan dan kini Malaysia telah mengambil inisiatif dalam memperkembangkan sektor permainan beranimasi ini.